1

National Library
of Canada

Bibliothèque nationale
du Canada

Canadian Theses Service

Services des thèses canadiennes

Ottawa, Canada
K1A 0N4

CANADIAN THESES

NOTICE

The quality of this microfiche is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Previously copyrighted materials (journal articles, published tests, etc.) are not filmed.

Reproduction in full or in part of this film is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30. Please read the authorization forms which accompany this thesis.

**THIS DISSERTATION
HAS BEEN MICROFILMED
EXACTLY AS RECEIVED**

NL 339 (1 86/01)

THÈSES CANADIENNES

AVIS

La qualité de cette microfiche dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

Les documents qui font déjà l'objet d'un droit d'auteur (articles de revue, examens publiés, etc.) ne sont pas microfilmés.

La reproduction, même partielle, de ce microfilm est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30. Veuillez prendre connaissance des formules d'autorisation qui accompagnent cette thèse.

**LA THÈSE A ÉTÉ
MICROFILMÉE TELLE QUE
NOUS L'AVONS REÇUE**

Canada

National Library
of Canada

Bibliothèque nationale
du Canada

0-315-22191-7

Canadian Theses Division

Division des thèses canadiennes

Ottawa, Canada
K1A 0N4

PERMISSION TO MICROFILM — AUTORISATION DE MICROFILMER

• Please print or type — Écrire en lettres moulées ou dactylographier

Full Name of Author — Nom complet de l'auteur

Pavel KALAE

Date of Birth — Date de naissance

June 25, 1958

Country of Birth — Lieu de naissance

Czechoslovakia

Permanent Address — Résidence fixe

2895 Oxtersong Dr.
Ottawa, Ontario
K1V 7B2

Title of Thesis — Titre de la thèse

COMPUTER USE IN STRUCTURAL ENGINEERING
WITH EMPHASIS ON PRE- AND POSTPROCESSING

University — Université

CARLETON UNIVERSITY

Degree for which thesis was presented — Grade pour lequel cette thèse fut présentée

MASTER OF CIVIL ENGINEERING

Year this degree conferred — Année d'obtention de ce grade

1984

Name of Supervisor — Nom du directeur de thèse

DR. GILBERT A. HARTLEY / DR. NEAL M. HOLTZ

Permission is hereby granted to the NATIONAL LIBRARY OF CANADA to microfilm this thesis and to lend or sell copies of the film.

The author reserves other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

L'autorisation est, par la présente, accordée à la BIBLIOTHÈQUE NATIONALE DU CANADA de microfilmer cette thèse et de prêter ou de vendre des exemplaires du film.

L'auteur se réserve les autres droits de publication, ni la thèse ni de longs extraits de celle-ci ne doivent être imprimés ou autrement reproduits sans l'autorisation écrite de l'auteur.

Date

June 28, 1984

Signature

Pavel Kalae

ML-01 (4/77)

**COMPUTER USE IN STRUCTURAL ENGINEERING
WITH EMPHASIS ON PRE- AND POSTPROCESSING**

by

© Pavel Kalab, B. Eng.

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of
Master of Civil Engineering

Faculty of Engineering
Carleton University
Ottawa, Ontario, Canada
June 25, 1984

Table of Contents

1	Introduction	1
2	Software	3
2.1	Application programs	3
2.2	Preprocessors	5
2.3	Postprocessors	7
3	Hardware	9
3.1	Mainframe computers	9
3.2	Minicomputers	11
3.3	Microcomputers	12
3.4	Communication	14
4	Computer use in the design office	16
4.1	Present applications	16
4.2	Future trends	18
5	Design of sample computer programs	21
5.1	Design concepts	22
5.2	Sample structure	22
5.3	Programming language	24
5.4	Logistics	26
6	General programming features	29
6.1	OPEN and CLOSE statements	29
6.2	Internal input/output and character manipulation	33
6.3	Branches and loops	34
6.4	Miscellaneous commands	34
7	Program description and use	36
7.1	Preprocessor	40
7.1.1	Initializing variables and opening files	40
7.1.2	Input of data	42
7.1.3	Editing of geometry data	43
7.1.4	Error checking and input of load data	44
7.1.5	Checking the aspect ratio	45
7.1.6	Calculation of geometry	46
7.1.7	Calculation of loads	50
7.2	Postprocessor	50
7.3	Translation program	52
8	Conclusions and recommendations	57
8.1	Conclusions	57
8.2	Recommendations	57

The undersigned recommend to the Faculty of Graduate Studies and Research acceptance of the thesis "Computer Use in Structural Engineering with Emphasis on Pre- and Postprocessing" submitted by Pavel Kalab, B. Eng., in partial fulfillment of the requirements for the degree of Master of Engineering.

G. A. Hunter

Thesis Supervisor

Mr. Selvadurai

Chairman,
Department of Civil Engineering

Carleton University

Ottawa

Abstract

This thesis examines the present state-of-the-art in computer software and hardware as it relates to computer-aided design in structural engineering. Particular emphasis has been placed on the use of pre- and postprocessor programs and engineering workstations to show how relatively small and inexpensive computers can and should be used. To illustrate some of the concepts of automatic preparation of data, transfer of data between computers, and automatic postprocessing of results, a preprocessor, postprocessor, and a translation program for coupled shear wall systems have been developed. The general features of FORTRAN 77 which facilitate the design and implementation of such programs have also been discussed. An attempt has been made to provide a document that could be used in the education of engineers who have only limited knowledge of computers, but who nevertheless extensively use or could use computers in their work, and a glossary of selected terms has been included to acquaint the reader with some of the jargon used in the field of computer-aided design.

Keywords:

computer-aided design, data transfer, hardware, preprocessors, postprocessors, software

Acknowledgements

I wish to express sincere gratitude to my supervisors, Professor G. A. Hartley and Professor N. M. Holtz, for their support and guidance throughout the course of this work, and for sharing with me their knowledge about computers and computer programming.

My sincere thanks are due to Mr. C. A. Carnegie and employees of the Regional Municipality of Ottawa-Carleton, Structures Branch, for giving me my first experience in pre- and postprocessing, thereby inspiring my research in this field, and to Mr. D. Godward and employees of Control Data Corporation for their assistance in running STARDYNE and transferring data to and from their mainframe computer.

I also wish to thank my friends, particularly Peter Slinn, Gordon Fenton, Mehmet Baltacioglu, and Rod Gillard for their numerous suggestions and comments.

Glossary of selected terms	59
References	63
Appendices	
A Program listing of preprocessor	A-1
B Program listing of postprocessor	B-1
C Program listing of translator	C-1
D Execution of preprocessor	D-1
E Input data for STARDYNE	E-1
F Output produced by STARDYNE	F-1
G Execution of postprocessor	G-1
H Output processed by postprocessor	H-1

List of figures

5-1	General network of pre- and postprocessors	23
5-2	Actual configuration of computers and how they were used	27
5-3	Ideal configuration of computers requiring access to service bureau software	28
6-1	Source code for subroutine OCHECK, which opens all files	32
7-1	Diagram of coupled shear wall system modelled by the preprocessor	37
7-2	Flowchart of the preprocessor operating in the normal data-entry mode	38
7-3	Summary of input parameters displayed by the preprocessor	39
7-4	Nodes and elements as they are numbered by the preprocessor	47
7-5	Simulation of the beam-wall connection	48
7-6	Flowchart of the postprocessor	49
7-7	Example of processed output	53
7-8	Example of unprocessed output	54
7-9	Location of element centroids as defined by the postprocessor	55
7-10	Sign convention for the element forces and stresses	56

Chapter 1

Introduction

Mankind has used various tools throughout history in the pursuit of his need for information and communication. The first records registered quantities of grain, animals, and possessions with sticks, pebbles, or marks on cave walls. This primitive method of counting was eventually replaced by a mechanical pebble-counter, the abacus, which in turn was replaced by a more advanced machine for mechanical calculations developed by Blaise Pascal in 1642 [16]. In 1822 Charles Babbage designed and built his first Difference Engine, which was a machine based on the same principles as today's electronic computers, and developed new ideas for a much more powerful tool, which, due to the limitations of the existing technology, could not be built. In 1890, Herman Hollerith [10] implemented the concept of punched card processing and formed a company to manufacture his equipment. In 1924 this company was renamed to International Business Machines (IBM), a name now well-known in the computer industry.

The first large scale automatic digital calculator, the Mark I, was completed in 1944. It consisted of seventy-eight devices linked by 500 miles of wiring and 3304 electromechanical relays. Mark I could perform twenty-three digit additions and subtractions in 0.3 seconds and could multiply two twenty-three digit numbers in about six seconds [16].

The first electronic computer, the ENIAC (Electronic Numerical Integrator and Calculator), was completed in 1946. ENIAC was so bulky it filled a large room, it used vacuum tubes which failed constantly, and could multiply two ten-digit numbers in 0.025 seconds.

With the introduction of the transistor, ferrite core, and printed circuits in the late 1950's, computer technology has advanced at a tremendous pace. Computers have become smaller, more powerful, more reliable, and much cheaper, so that they are easily affordable not only by large research and educational institutions, but by small consulting firms and individuals as well [6]. This advance in technology has had a significant impact on the engineering profession. The structures being built today are generally so large and complex, that their design by methods predating the development of the computer is either nonfeasible or uncompetitive. For this reason, computer-aided design (CAD) has evolved as a special branch of most engineering disciplines.

The computer hardware industry has recently entered a new phase of growth, which presents potential for drastic changes in the way that structural engineers carry out their work. However, the software industry has not kept pace with these developments so that a gap exists between the state-of-the-art hardware and the software making use of it. Fortunately, this gap has the potential of being reduced by graduates of civil engineering who are more knowledgeable in the use and capabilities of present-day computers, and graduates of the school of computer science, who are skilled at translating engineering concepts into computer source code.

The author did not intend this document to be of interest to computer specialists, but it is directed at practicing engineers, who have only limited knowledge of computers beyond programming in the FORTRAN language, but who use, or could use, computers to make their work more efficient. This thesis is intended to acquaint structural engineers with the advantages of using pre- and postprocessors in computer-aided design by:

1. Examining the state-of-the-art software and hardware available at the present time, considering some of the future trends of the computer industry, and discussing how computers may be used in the engineering design office.
2. Describing what pre- and postprocessors are, how they may be written to take full advantage of a programming language such as FORTRAN 77, and how they are used.
3. Illustrating the above points with a specific example.
4. Providing a limited glossary of some of the terms used in computer-aided design.

Chapter 2

Software

When computers were first used for engineering analysis and design, few programs were available, and engineers wrote programs to solve their own particular problems. These programs were limited in application and were available only to the company or person that developed them. However, there are now many companies which specialize in the development of computer programs (software) for a number of applications which can be executed on a variety of computers ranging from the large mainframes to the small microcomputers.

2.1 Application programs

Commercially available software for engineering applications can be divided into three distinct groups. The first of these is the general-purpose program, such as a program which is used for finite element or frame analysis. The second type of program is a specific application program, which may be used to analyse only one type of structure. The third type of program is a front- or rear-end (also known as a preprocessor and postprocessor, respectively), which is used in conjunction with a general-purpose program to give the appearance to the user that a specific application program is being executed.

The large general-purpose programs make use of the finite element method of stress analysis or matrix frame analysis. These methods can be used for problems ranging from simple flat plates to complex structures such as bridges, dams, or nuclear reactors. Most have special features which the engineer may want to use, such as dynamic analysis, nonlinear geometrical properties, inelastic material properties, and thermal analysis. This technique has permitted stress analysts to solve problems, which could not have been handled by classical elasticity methods, or problems, where the calculations are so numerous and complex that they could not be solved without the computer.

Such programs are usually maintained and operated by a commercial service bureau. The various programs and their capabilities and limitations have already been extensively discussed in literature [8, 13, 17, 18, 19, 34, 35, 36]. Since these programs are designed to handle a large number of diverse problems, the input and output of data is frequently complex and difficult to understand.

The problem definition often requires extensive information, usually entered in fixed format, and the program may be capable of executing only on large computers due to the requirements for a large memory.

For this reason, the trend in recent years has been to develop specific application programs, which can be used to analyse only one type of structure. The advantage of using such programs is that the input of data is simplified considerably, because the user needs to input only a few parameters to define the structure in general terms, and the program automatically calculates any other data, which may be required, from these few parameters. Also the output is tailored to the specific needs of the designer, and data which may not be shown by a general-purpose program are displayed, or are displayed in such a format that the output can be directly incorporated into an engineering report. Hence the specific application programs are more convenient to use than the general-purpose packages.

The other major advantage of specific application programs is that they are generally small enough that they may be executed on a mini or microcomputer. Already numerous software developers have tackled this large market and made available engineering software, which, for example, is used to design foundations, beams, retaining walls, etc. on machines using the CP/M or MS-DOS operating systems, i.e. the small personal computers which cost approximately \$5,000 [17].

An extension of this concept is the linkage of a specific preprocessor, a general-purpose program, and a specific postprocessor to simulate a specific application program. The user has the convenience of using a program which requires minimal input data, since the preprocessor calculates most of the parameters required by the general-purpose program. He also has the full computing power of a large program, and the convenience of easily interpreted output once it is processed by the postprocessor. Pre- and postprocessors are relatively easy to develop for different structures, so that they may be developed in-house by the engineers who will eventually be using them. The engineering firm may therefore find it convenient to use one general-purpose program with one pre- and postprocessor couple for each structure that is frequently analysed, rather than using several specific application programs.

When looking at the use of computers in a design office, one must always consider whether computer analysis is cost effective. The cost of computer analysis may be broken down as approximately 70% of the analysis cost for problem definition and preparation of input data, 20% for the interpretation of the results, and only 10% for the actual computer input/output and calculations [40]. Furthermore, when connected to a service bureau computer, the greatest component of the computer cost is often associated with the on-line printout of data. Hence, it is imperative that

the cost of data preparation, input, and interpretation be reduced by minimizing the connect time to service bureau computers. This can be done only with the use of pre- and postprocessors which can be run on smaller computers directly in the design office.

2.2 Preprocessors

Preprocessors are programs which have either been integrated into the application program, or which are independent units, and which generate the input data used by the general-purpose application programs. An ideal preprocessor is one that allows the engineer to generate the necessary information with the least effort for as wide a range of problems as possible within the category covered by the preprocessor, an ideal which is very difficult to achieve in practice. Furthermore, the preprocessor should be "user-friendly", flexible so that the designer may follow any logical path he wishes in defining a structure, allow previous status recovery in the event that the user makes an input error, have free format input, visual feedback, on-line documentation, expandability, and transportability to other computer systems [35].

The preprocessor is the user's window to the computer and the target program. It is not algorithmic, but serves only as an interface, and should be designed completely with the user in mind. If the ordering or arrangement of data to the preprocessor is done basically for a convenient transition into the algorithmic program, the preprocessor will be an inferior one. It is likely that good preprocessors would be designed by applications engineers who know nothing about the target program (which might prejudice their views or design). It is for this reason that a good preprocessor may in fact be much longer and more complex than the target program. Many computer users will be of the opinion that the best code is the most efficient. With the cost of computing at an all-time low, and decreasing constantly, a new philosophy that is gaining momentum is that a good program is one which is convenient for the user.

There are several types of general-purpose preprocessors. Probably the most simple is a translation program which accepts data in free format and translates it into the format required by the application program. Only a few lines of code are needed for this type of preprocessor, and hence it can be written by people with very limited programming experience. It may expedite the input of data and avoid errors associated with missing decimal points and faulty right justifications. However, the usefulness of such programs is relatively limited.

An improvement on this type of preprocessor is one which is run interactively from a video display terminal (VDT) and prompts the user to enter data as required. Although more difficult to write, there are a number of advantages to this type of program. The use of well-designed conversational input can eliminate the need for a user's manual so that the program is autoexplanatory.

At the same time as data are entered, checks can be made, and the user warned if the input exceeds expected limits [46].

Computer graphics can provide another possible improvement in preprocessor design. Again, the simplest is a translation program which enables the user to translate the information contained in the data file for an application program into a format which is accepted by a general-purpose graphics package. The graphics program allows the user to manipulate an object so that by using a combination of various translation, rotation, and magnification commands, the design is viewed from various perspectives [9]. The benefit in being able to scan and check extensive data is significant. However, such a program is limited because it only allows the user to look at information stored in a data file, but does not allow any interaction which would modify that data.

A better approach is to use an interactive graphics package which allows the user to generate a data file by using an on-line graphics terminal. Several such preprocessors, which allow input data to be created, edited, and reviewed at a graphics terminal in timesharing mode, are in use [12,15]. A node number and bandwidth optimizer as well as an automatic mesh generator and automatic 3-D curve generator may be included, and coordinates may be entered in the Cartesian, cylindrical, or spherical coordinate system.

However the power and efficiency of preprocessing cannot be fully realized in general-purpose preprocessors. In order to cover as many cases as possible, the preprocessor programs have many features which may make their use complicated. Hence instead of learning the complicated instructions needed to run an application program, the user must now learn the complex commands needed to operate the preprocessor efficiently. This appears to defeat the purpose of preprocessors, whose primary function is to simplify computer use. Most structural designers and structural design firms tend to have a few specialties (or generic structure types) which are designed over and over again. The differences from one design to another can be described by a few of the design parameters. In such cases, it is more economical to use very specific preprocessors suitable only for this type of structure, since very large data files can be generated when only a few parameters are entered [27,28].

It is the author's view, that in the particular case considered in this thesis, the preprocessing of data for large-scale general-purpose frame and finite element programs, this is the only aspect of preprocessing worthy of serious consideration. In many consulting firms which regularly develop and process large structural models on the computer, the engineer/analyst codes the data on a coding form for later verified keypunching by a keypunch operator at a data processing center. Most of these data will consist of spatial coordinates of nodes, member identification and properties, and load cases. All of this information is in tabular form. A table in free format cannot be verified

very easily, and the engineer will be required to check the listing in a reasonably short time. In a rigidly formatted file listing, even missing decimals are readily found under minimal scrutiny. The cost of a professionally keypunched file is the same whether it is in free or fixed format. The only worthwhile aim in preprocessing to this engineer/analyst is to reduce the several thousand lines of input by at least an order of magnitude. In specific applications, this can be easily done, and more thought and development work devoted to this is long overdue.

With the recent developments in computer workstations of a very high level of sophistication but limited capacity (compared to a service bureau), the philosophy of preprocessor development takes on a new significance and urgency. The workstation can be used to solve many of the smaller sizes of applications of the design office, but more significant to this thesis, it can be used to preprocess data files which will ultimately be run on a service bureau computer.

2.3 Postprocessors

The postprocessors, like the preprocessors, range from very general to very specific. Most application programs have provisions to list maximum and minimum values of forces, stresses, and displacements, to flag values falling in a previously specified range, and to plot various output data. In most cases, the output format must be specified when the data are prepared, and the information is merely dumped to the appropriate output device [14].

The disadvantage with such an approach is that the postprocessing parameters must be specified before the problem is analysed, and this does not provide flexibility. If the user wishes to examine a different range of values, he must either look at the unprocessed output data, or run the job again with different processing commands. This is inefficient.

Therefore, general and specific postprocessors have been developed to allow the user to interactively examine selected regions of the output. The use of graphics is very important as it facilitates interpretation of data, and is much easier to understand than extensive tables of numbers. In this way, postprocessing enables the user to access the output data file of an application program, and, for example, draw the deformed shape under a variety of loading combinations, contour plots of the stresses or temperatures in a structure, shear and moment envelopes of selected areas, various cross-sections through the undeformed and deformed structure, natural vibration mode shapes, and generate the time history of the structure. This information can be transferred from a graphics screen to a plotter so that the hard copy can be used in a report. As with the preprocessors, if only a certain type of structure is analysed, it is more economical to use a specific postprocessor instead of a general one, because it is easier to use and its full capability can be utilized.

Therefore, two specific styles of post-processing are addressed here. Both of them are especially important if the output file is large. One of these postprocessors is referred to as a general-purpose postprocessor, and permits a great deal of flexibility in the selective printout from a large data file consisting of a series of tables. The other type of postprocessor is the specific postprocessor, and it is unique for each structural application. For example, a number of applications of the finite element method are not practical, because the output file from a general-purpose program either does not contain the information that the engineer requires for design purposes, or buries this information in a very large volume of useless data. In such cases, postprocessors are invaluable tools which the engineer uses to extract the information already present, or to calculate new parameters based on the data found in the output from the application program.

Chapter 3

Hardware

Just as software was developed from very awkward and inefficient algorithms to fast, easy-to-use programs, the hardware has also undergone a considerable transformation. Some of the earlier computers were no more powerful than some of the modern hand-held programmable calculators, yet occupied room-sized areas, consumed a great deal of power, and broke down so frequently that running programs considered today to be of only moderate size would not have been possible. The development of microcomputers such as the IBM PC, APPLE, HP, and their numerous contemporaries heralds a revolution in the computer industry, and makes available vast and inexpensive computing power for engineers and laymen alike.

3.1 Mainframe computers

The most powerful of a long line of machines is the present mainframe computer. It generally has a relatively large wordsize of 32 to 64 bits. The "word" is a basic addressable unit such as a real number, an integer, or a character string. The larger wordsize implies that calculations can be done more quickly and accurately, and a larger core memory may be addressed.

However, the speed, accuracy, and the memory size which may be addressed depends on the internal architecture of the microprocessor, and considerable confusion exists when one tries to define which chip is "better" or "faster". Generally speaking, a 36-bit mainframe such as the Honeywell CP-6 at Carleton University can use integer numbers in the range from -2^{35} to $2^{35} - 1$, and floating point numbers with a single precision of 7 significant digits and a double precision of 18 significant digits. An 8-bit microprocessor can represent integers in the range of -2^{15} to $2^{15} - 1$, since the 8-bit processor has some double registers. Single precision numbers usually have seven digits and double precision numbers have 14 significant digits. However, under software control, the accuracy (i.e. the number of significant digits) can be increased or decreased, so that the capabilities of the microprocessor become meaningless. For example, the capabilities of the Intel 8088 used by the IBM PC can be enhanced by using the appropriate software to provide 18 digit accuracy. But the software was very slow, and Intel designed the 8087 mathematics co-processor which increases the numerical accuracy and speeds up calculations. Some characteristics which

are used to compare microprocessors are the frequency at which they operate, and the sizes of the registers, arithmetic logic unit, internal data path, external data path, real memory address, virtual memory capabilities, etc. However, microprocessor architecture is an extensive field in itself and it is beyond the scope of this thesis to examine it in greater detail.

The mainframe computers are large (both with regard to physical dimensions and the memory used), fast, and expensive, and for these reasons they are operated mainly by computer service bureaux, educational institutions, government research laboratories, and large consulting firms, which have enough computer work to make such an investment cost-effective.

Mainframe computers are multi-user, which means that a number of users can have access to one machine at any time. Interaction with the computer takes place through any number of input/output devices, the most common of which are the visual display or hardcopy terminals. Cardpunching, which was once the principal mode of input, has almost become obsolete. Although cards are available from a few keypunch service outlets, the IBM 029 cardpunch is no longer manufactured, and this service will completely disappear in a few years. The terminal may be located in close proximity to the computer so that data transfer takes place through hard-wired connections at very fast rates, or the terminal may be operated from a remote site such that data is transferred via telephone lines at a much slower rate.

Only very few large firms would find it economical to purchase a mainframe for in-house use. Such a computer may provide lower actual computing time costs, the convenience of rapid turnaround, and the possibility of developing large-scale proprietary software. In addition, some of the operating costs may be recovered by renting computer time to outside users. There are several disadvantages, however, which must also be considered. User-support personnel and computer support staff must be provided by the firm. These are computer specialists, and not generally engineering personnel. Computer servicing contracts are expenses that must be met, including not only repairs, but preventive maintenance as well. For this reason, many firms prefer to make a smaller investment by purchasing a minicomputer or a microcomputer, and by using the service bureau's mainframe for jobs which the in-house computer cannot handle. It is this integration of a remote mainframe with a local computer workstation, and the further integration of remote and powerful general-purpose software adapted to local needs using the local facility, which could well be a major factor in how we will make use of computing in civil engineering in the future.

It is perhaps also appropriate to point out that many of these applications of the future can be implemented by programmers (persons who have received special training in a 3-year college program), since these developments are not algorithmic or technical, but merely manipulate and format data. It will then be our job as engineers to supervise and oversee these developments.

3.2 Minicomputers

Minicomputers are scaled down versions of the larger mainframe computers. They are not so powerful since they usually have smaller word sizes, which means lower precision and access to a smaller core memory. The execution time is longer, but the cost (both initial and operating cost) is much lower than for a mainframe.

Minicomputers are generally dedicated systems used by one person at a time, although they can be used with multi-user operating systems in a timesharing mode. Due to memory restrictions, large application programs which require 64-bit word sizes sometimes cannot be run on these smaller machines. However, large programs with a high demand for core storage can often be made to "fit" a minicomputer by one or more methods. For example, virtual memory is a facility available in many computers which enables "pages" of data stored in secondary memory such as a floppy or hard disk to be swapped with data in core memory. In this way, the computer pretends that it has a much larger core memory than it in fact has, so that some very large programs can be run on relatively small and inexpensive computers. For example, a static and dynamic analysis of a 20-storey building (230 nodes, 460 members, and 660 d.o.f.) was performed on a 32-bit VAX 11/780 using the SAP VI analysis program [21]. More recently, work is being carried out to transfer large finite element programs, solids modelers, and various graphics programs, which until now have been running on a large mainframe computer at Carleton University, to the Apollo workstations, which are based on the Motorola 68010 microprocessor, and which cost \$35,000 as compared to the multi-million dollar Honeywell mainframe computer.

Programs can often be segmented into separate modules which are run independently, and which write data to disk files for use by subsequent modules. If the modules themselves are given "user-friendly" names such as INPUT, ANALYSE, etc., the segmented program need be little more complicated than a single run. If it is inconvenient to segment a program, then it may be possible to segment the structure to be analysed. This procedure is called substructuring, and can be advantageously applied to frame and finite element analysis. The basic concept is that only a small segment of a large structure is modeled so that the boundaries, where the segment is connected to the rest of the structure, are completely restrained from any movement. The analysis will indicate that "fixed-end" forces are present at these boundaries, and the forces present are then used as "applied forces" when the neighbouring segment is analysed. In this way, a large problem may be broken down into several components which can be readily analysed on a small computer even though the entire structure could not be processed.

Although minicomputers can be used for structural analysis using interactive application programs, they are particularly well-suited to be used for pre- and postprocessing. The general

graphics packages may be too large to run on such a system, but specific non-graphical pre- and postprocessors may be used. Also, the computer can be used with communications software and a modem, so that it acts as a terminal to a larger mainframe computer with a small but important difference. Data can be transferred back and forth as completed files, so that the printing of input and output data is actually done off-line (i.e. unconnected to a mainframe).

3.3 Microcomputers

A microcomputer can be defined as an electronic computer which has its CPU on a single microprocessor in contrast to the minicomputers and mainframes, which use several discrete components. The advances made in the field of microcomputers are so rapid, and so much literature has been devoted to this topic, that it is impossible to cover here in any great extent. Publications such as Byte, IBM PC, PC World, Microcomputing, Interface Age, and many more announce new hardware and software developments in every issue.

Initially microcomputers started as machines with only 8-bit words. They were based mainly on the Zilog Z80, Motorola 6800, and other proprietary microprocessors. Although the microcomputers were small and slow, they were relatively inexpensive (\$2,000 to \$10,000) so that they were readily available to engineers. Even though they were initially looked upon as toys (in fact the first microcomputers were used for playing arcade games rather than numerical calculations), their full engineering potential was emphasized when a past director of NASA commented that "the entire Mercury Program was designed on a computer which was slower and had less memory than the APPLE" [41].

Almost at the same time as the 8-bit systems such as the APPLE, Orion, Commodore, and many others made their appearance, IBM announced its 16-bit microcomputer, which was to set new standards for the industry. This system consisted of the Intel 8088 microprocessor, a high-resolution graphics monitor, internal memory which could be expanded to 640 K bytes, two 5 $\frac{1}{4}$ inch floppy disk drives capable of storing 360 K bytes each, or a hard disk which could replace one of the floppy disks and store from 5 to 50 M bytes, and the options to add many more peripheral devices (printer, plotter, modem, etc.). The computer industry set this machine as a standard and there was soon a proliferation of IBM-compatible hardware and software. Some very large programs could now be run on these inexpensive machines, and it was soon realized that the microcomputers would eventually rival the minicomputers and the mainframes as well.

But the industry did not stop with 16-bit designs. By the end of 1984, 32-bit microprocessors will be manufactured in commercial quantities by DEC, HP, Inmos, Intel, Motorola, National Semiconductor, Zilog, and other companies. Already Hewlett-Packard is marketing the HP-9000,

Table 3-1: Summary of computer types

TYPE	COST IN DOLLARS	WORDSIZE	LANGUAGE
Programable calculator	50 to 400	—	No standard
Desktop computer	2,000 to 10,000	8 bit	Basic
Microcomputer	3,000 to 20,000	16 bit	Basic, Fortran Pascal, etc.
Minicomputer	10,000 to 500,000	32 bit	same
Mainframe	0.5 to 5 million	64 bit	Most languages

Table 3-2: Relative computer speeds (operations per second)

COMPUTER	REAL	LOGICAL
Microcomputer	1,000	100,000
Minicomputer	10,000	200,000
Super mini	100,000	1,000,000
Mainframe	1,000,000	5,000,000
Real operation: $A = B + C * D$; Logical operation: IF (I.E.Q.J) ...		

From: Wilson, E. L.; The use of minicomputers in structural analysis. Computers and Structures, Vol. 12, p. 696, 1980.

which is a 32-bit workstation, and DEC plans to introduce its Micro VAX 1 with a microcomputer based on the same architecture as its successful line of VAX 11 minicomputers [31].

The importance of such systems for engineering applications cannot be underestimated. The microcomputers are now large enough to operate in most of the high-level programming languages such as BASIC, FORTRAN 77, PASCAL, C, LISP, FORTH, and others, which, until recently, could not be completely implemented. The high resolution graphics and large core and secondary memories enable engineers to run large CAD programs, and the extensive use of such systems ensures a wide software base. The microcomputers can be used in multi-user networks, so that for most applications it is no longer necessary to rent computer resources from a service bureau. As new advances are made in this field, the microcomputers will become even more powerful and less expensive, so that design firms will be buying 64-bit machines to replace their 16-bit contemporaries. The different types of computers presently available and their characteristics are summarized in Tables 3-1 and 3-2.

3.4 Communication

The ability to transfer data from one machine to another has become very important. As early as the 1950's it was evident that computers would need to talk to each other, but only in the late 60's and early 70's were formal mechanisms defined to provide data communication. As computer systems expanded and users became aware of the potential of remote data processing, the need for better and faster communication became apparent.

There are three methods by which data may be transferred. The first is the simplex, or one-way communication, which permits the flow of data only in one direction. The second is the half-duplex, which allows two-way communication, but only one way at a time. The disadvantage with the simplex method is that although transmission is fast, two lines are required, one for each direction. The disadvantage with the half-duplex method is the long response time, since transmission cannot take place while data is being received. Therefore, there is the third method, the full-duplex, which allows simultaneous communication in both directions.

Most information today is transmitted using the ASCII code, which uses seven data bits per character and an added parity bit to provide error checking. These digital bits are converted to analog form so that the information can be transmitted over the established telephone network. A two-frequency modulation technique was devised with one frequency representing the 0 and the other representing the 1 of the binary code. The modulated signals require a device which modulates the signal at the transmitting end and demodulates it at the receiving end, a device which is appropriately called the "modem".

In deciding how to communicate between remote locations, the user has a number of alternatives depending on his requirements for transmission speed, data channel availability, reliability, and cost. A user may connect to the dial-up telephone network via modems, or have a dedicated data route installed. Data transmission rates may be as low as 150 baud (approximately equal to 150 bits per second, which is now abbreviated 150 bps), or as high as 1.5 M bytes per second using high-transmission schemes. For practical engineering purposes, data transmission rates are 300, 1200, or 2400 baud over long distances, and 2400 to 9600 baud over short distances. Hard-wired connections are the most cost-effective as long as the distances are short, because signal degradation and external noise have little effect on data transmission, and hence transmission rates can be very high. Unfortunately, there is a finite distance over which the signal may be transmitted in this way due to resistance in the wire, and for long distances, the rates must be much slower. The problem with using a slow rate (e.g. 300 baud) is that it takes a long time to transfer data; for example, a 60-page printout would take approximately 5 hours to print. This means that the telephone line is occupied for a very long time, and the connect costs to a service bureau would

be very high. Furthermore, there is a high probability that the computer or the line may crash during such a long time interval, and there may be a considerable amount of noise on the line, which may make the data unintelligible, or may introduce fatal errors which are not readily apparent. Transmitting at 1200 or 2400 baud increases the cost, but reduces the risk of crashes and noise interference, so that it may be more economical in the long-run [4]. As a rough guide, a regular user of a computer terminal on-line (i.e. interacting directly with the computer) would find that 150 baud is intolerably slow, 300 baud is acceptable for keypunching files and printing short files, but slow for long ones; 1200 baud is fast for a hardcopy terminal and acceptable on a video terminal, but slow for graphics; 2400 baud and higher is very expensive and is appropriate for interactive graphics. If large quantities of data must be transferred over long distances, it may be more economical to use punched cards or magnetic tape rather than on-line transmission.

In addition to the hardware required so that computers may talk to each other, special software is also needed. This communications software enables the transmission and capture of ASCII and binary data files and permits the computer to act as a terminal to another host system.

The use of microcomputers and data communication over public telephone lines means that engineers can be physically distant from the design office, yet still have access to all the necessary information. Datapac is one such network which enables the user to access any computer system connected to it by dialing a local Datapac telephone number and attaching his computer as a remote terminal. In addition to allowing users to simply connect to another computer, Datapac also provides electronic mail between remote sites and allows office text machines from different manufacturers to communicate with each other. Service such as this is particularly important if an engineer is working in the field and needs to perform an analysis immediately on the spot. It is also a convenience for those who prefer to work at home rather than commute from the suburbs to the downtown office every day. This is already happening to a large extent in the Silicon Valley, and to a smaller extent in Ottawa, where companies such as Bell Northern Research, Mitel, ComDev, etc. provide home computers for their employees so that whenever they get an idea, they can pursue it without any delays. This approach may affect our lifestyles in a similar manner as the automobile, and release designers from a rigid working environment.

Chapter 4

Computer use in the design office

4.1 Present applications

Numerical calculations are used in most aspects of civil engineering. In structural engineering, computers are used to design small components of structures such as beams, slabs, or columns, or entire structures such as bridges, dams, and buildings. In geotechnical engineering, computers are used to simulate soil-structure interaction, design footings or retaining walls, and analyse slope stability. In municipal engineering, extensive calculations are required to analyse water distribution and sewage networks. Transportation engineers use computers to calculate cut and fill quantities for roads, carry out discrete simulations of traffic flow patterns, or find the optimum route for a transportation line.

One reason why computers are such an important tool for civil engineers is that various software packages have been developed which solve almost any design or analysis problem. "Off the shelf" computer programs, whether those supported by a service bureau, or those bought for the office microcomputer, have been developed by computer specialists using the latest state-of-the-art techniques so that the programs are efficient. The programs have been rigorously tested to ensure that they work correctly, and if problems do occur in recently developed programs, they are usually fully supported, meaning that repairs will be done quickly if required. The programs are usually interactive, provide on-line documentation, or are accompanied by detailed manuals. This means that the user requires little or no computer training (aside from a university engineering education) to operate a computer program.

As structures become more complex, engineers are asked to consider more parameters and evaluate several alternatives before arriving at the optimal design. Computers make this a relatively easy task because time-consuming calculations are performed very quickly so that the engineer's creative output increases. If the firm owns a computer, the engineer might be inclined to do a more rigorous computer analysis than if a service bureau computer was used, because the general philosophy is "as long as the machine is sitting here, we may as well run another simulation". Thus in-house computing is not necessarily cheaper, but it does lead to more computing and therefore a more comprehensive engineering analysis.

There are other advantages to using computers. The stored data is much more compact and hence more of it can be stored in a smaller physical volume (i.e. magnetic tapes, hard disks, and floppy disks). In most cases, such large quantities of data would be impractical to store and handle by any other means than with the computer. Input, retrieval, and updating is also much easier. For example, in mapping and surveying, raw data can be recorded digitally in the field using various data acquisition systems, and this data may then be transferred to an office computer where it is processed automatically into maps [43]. Raw data may also be obtained from satellite or aerial photographs, converted to two- and three-dimensional maps, and utility or transportation routes may be plotted directly by the computer according to predefined criteria.

Another important application of computers is in the creation and use of databases. The need for handling large quantities of data developed concurrently with major applications. There was a need for storing constants such as material properties, dimensions of steel sections, etc. for various analysis and design programs. If this information was stored in files which were separate from the application program, the data could be used by multiple projects using the application, and still be extended or modified as required. When changes were made to the data, they applied to all projects using the program. But as the number of application programs has grown, the generation and management of input data has become problematical, because each file of input data has a separate structure. Data preparation and the writing of interfaces between file formats has become a major endeavor of many engineering groups and has led to extensive use of pre- and postprocessors. It was soon realized that data common to a number of applications need be entered only once and used as input for all applications with reformatting taking place as needed on input and output. Such a common repository has come to be known as an integrated design database, or simply a design database.

This type of integrated approach can greatly reduce the cost of writing a preprocessor for yet another application or for implementing yet another set of file transfer and mapping routines. Some of the benefits of using databases include the automatic generation of production data, such as drawings and numerical control machine tapes, or for generating summary reports. By keeping all design data in a common machine-readable form, the possibility exists to check consistency of data and thereby reduce or eliminate conflicts and improve the control over the design product. Since the data are stored in a ready-to-use format, databases support further automation so that a design database becomes an environment in which designers work directly, taking input data from the database and writing their results back to the database which is thus automatically modified and updated.

Hence it may be seen that at the present time, computers are valuable tools for engineers because they assist in the gathering of data, in its processing (by creating input data files, performing numerical calculations, and interpreting the output), and in its presentation (by using graphics techniques for generating charts and graphs, and high-quality printers for text processing) to carry out economical analyses and designs, and present the work in an attractive and professional document.

4.2 Future trends

Considering the advances in computer technology over the past few decades, it is difficult to predict where we will be toward the end of the century. The trend in miniaturization will continue, so that more computing power will be packed into smaller units. Laser optical disk technology will replace our present storage media so that external memory for microcomputers will be measured in gigabytes rather than kilobytes [20]. The graphics resolution of monitors will improve, and colour graphics will become a standard feature for all computers. The speed and capability of future microprocessors will increase, while the price of the entire system will decrease. It is possible that the pocket calculator of tomorrow will be as powerful as the minicomputer is today. Only a few years ago, this statement would have seemed ridiculous.

One area where significant advances will be made is in graphics and its applications. Although computer graphics are already being used extensively in civil engineering [44], there is considerable room for improvement. CAD systems of the future will make use of instantaneous displays of 3-D meshes, zooming, translation, and rotation without redrawing, partial image updating, blinking to highlight certain aspects, colour contouring, and many other features. This will eventually lead to extended use of solids modelling where "stick drawings" are used at the present time.

Although solids modelling is already in use in other fields such as mechanical engineering, it has not yet found significant application in civil engineering although it could be very valuable in design and analysis. One of the oldest and most persistent problems of designing complex projects is physical interference, that is, a condition where two or more objects shown in various drawings occupy the same physical space. From the design viewpoint, physical interference is particularly vexing to discover by manual techniques, and from the construction viewpoint, it is expensive in time and resources to correct. Therefore, solids modelling could be used to discover interferences before construction starts. The check is more thorough than manual methods, and it is a mathematical certainty that all interferences in the model will be found [5]. A particularly useful application of solids modelling is geotechnical engineering, where the simulation and drawing

of underground structures and the calculation of quantities to be excavated would be greatly simplified

Computer interaction is another area where significant advances will be made. Extensive research and development is being carried out in the field of speech recognition and voice-actuated machines [23,30,42]. At the present time, it is not yet possible to efficiently interact with the computer, by voice alone, although some experimental computers can now respond to a few voice commands. The main problems lie in the computer's limited ability to cope with different pronunciations, accents, and background noise. Even a limited vocabulary which the computer must recognize to execute a selected number of commands requires a large memory, which makes voice recognition impractical in present computers. Once these problems are overcome, interaction with computers (high-level programming and data input) will take place by natural speech rather than via the keyboard.

Closely associated with this is the idea of artificial intelligence and expert systems. Most interactive computer programs of today use some sort of error detection schemes and aids to help the user when he encounters problems. Furthermore, it is now possible for the computer to write its own programs, prepare documentation, and debug other programs [1]. However, this is not artificial intelligence as such, because it is static and unable to respond to the results that the program produces. Each execution is the same and the program does not improve itself.

The key feature of an intelligent program is the ability to self-modify its operation based on previous successes and failures. For the engineering design model, this implies the implementation of a feedback mechanism with the three main steps of intelligent design, namely the application of previous design experience, the extraction of appropriate information to be used according to that experience, and the storage of the modified design experience [37]. These objectives are achieved by combining data acquisition, information retrieval, searching, pattern recognition, and database manipulation. The computer thus learns as it progresses from solving very simple tasks to tasks which are more and more complex.

Reasonable success has been achieved with the so-called expert systems in medical diagnostics and prospecting. The possibility of applying this technology to civil engineering also exists. Most computer programs in use today attempt to automate the design and analysis of engineering projects as much as possible. Attempts have been made to combine several programs into one package, so that the user may, for example, generate the preliminary dimensions of structural members, calculate the loads, perform the analysis, estimate the costs involved, and plot the structural layout of the designed structure [26]. Although such programs can be very complex, they still rely on the designer to input specific data and do the "thinking", that is, these programs

cannot base a design on previous experience as the engineer can. Totally automated design is not presently possible, because there can be no mechanical substitute for creativity, however the possibility of using the computer for more complex decision-making functions still exists.

The Japanese have already embarked on a national enterprise to develop fifth generation computers which will use many of the features described here, as well as others which could not be covered. At the present time, we are in the middle of an electronic revolution, and when one considers what progress has been made in computer technology in the last 40 years, we can only speculate what the state-of-the-art will be another 40 years from now.

Chapter 5

Design of sample computer programs

Three computer programs have been written to illustrate the concepts discussed in the preceding chapters. They are a preprocessor which generates the data file required for analysing a coupled shear wall system using the STARDYNE finite element program, a postprocessor which interactively displays selected regions of the output from the FEM analysis, and a translation program for the graphics program MOVIE [9] which creates the required data file so that the geometry of the shear walls may be viewed on a graphics terminal. All programs have been written in ANSI FORTRAN 77 on the Carleton University Honeywell CP6 Computer System.

MOVIE is a commercial general-purpose graphics program which facilitates the display of three-dimensional mathematical models as line drawings. The program is capable of removing hidden lines, drawing contour lines, and displaying the drawings as black and white or colour shaded images. MOVIE reads a data file where the coordinates of all line intersections and the connectivity information about these lines is stored and then displays a graphical image of the object described in this manner. The user can rotate the object about the three coordinate axes, translate the object anywhere in space, zoom in on details, and "explode" the object so that it is easy to see how the various elements are connected. These functions may be combined to display a series of frames in a very primitive discrete animation. If displacement data about the various points of the structure are available, the data file may be read by MOVIE to display the "deformed" structure. Although MOVIE is not the best graphics package available on the market, and although it cannot be used to alter the geometry data of the structure, it is nevertheless useful as a check to ensure that geometry data have been correctly entered in the computer.

STARDYNE is a finite element static and dynamic structural analysis program. A static analysis will predict stresses and deflections resulting from pressure, temperature, concentrated forces, and enforced displacements. The dynamic analysis will predict the node displacements, velocities, accelerations, element forces and stresses from transient, harmonic, random, or shock excitations. STARDYNE is user-oriented and contains automatic node and element generation features that reduce the effort required to generate input data, but these features are not as easy to use as a true preprocessor, and require considerable knowledge about the structure to be modeled and at least some knowledge about how a finite element program operates. The program itself is

very "forgiving" and contains several error-detection routines which correct many errors that the user may have made. The output includes displacements, stresses, and forces, and also enables the user to make plots of the original model and the deformed structural shapes, and stress contours of two-dimensional elements. It is one of the more popular finite element programs which is supported by many service bureaux, and for this reason it was chosen as the main application program to demonstrate how pre- and postprocessors simplify structural analysis using finite elements.

5.1 Design concepts

The preparation and interpretation of data for finite element programs is long and tedious. The process can be greatly simplified by using general purpose pre- and postprocessors, but if only one type of structure is analysed, this would not be very economical because such programs would still require large amounts of unnecessary data to be entered, and their operation would be expensive. A more economical approach is to use specific pre- and postprocessors which require a minimal number of input parameters to generate all of the other data. Although such programs may require some effort during the development and testing stages, this disadvantage is outweighed by their simple operation and by the fact that such programs may usually be run on small machines (minicomputers or microcomputers), whereas the commercial programs generally require a service bureau mainframe with graphics capabilities.

In contrast to the general pre- and postprocessors, which are sometimes a single program which is used for both functions, the specific pre- and postprocessors require two independent programs which are linked as shown in Fig. 5.1. The preprocessor generates the input data file, which can then be accessed by the application program such as STARDYNE, or by other preprocessors which rearrange the data into a different format used by other application programs, such as the graphics program MOVIE. The application program performs the required calculations and generates an output data file. Postprocessors are then used to access the output and change it to a format more acceptable to the user. The pre- and postprocessors may be chained together to form extensive networks in which the data created by one program are translated to a different format and are used as input for another program. This may be repeated as many times as required.

5.2 Sample structure

The structure which was selected for illustration purposes was a coupled shear wall system. Such an example is practical because many high-rise buildings use shear walls to resist lateral wind and earthquake forces, it is simple to model because of its planar nature and regular geometry,

Figure 5-1: General network of pre- and postprocessors.

it is awkward to manually generate the data files because the number of nodes and elements required is large, few parameters are required to completely define the structure, and the analysis is complicated or not as accurate if performed by other means. Hence the coupled shear wall system is ideally suited for modelling using preprocessors.

Contrary to popular belief, finite element analysis is not necessarily more expensive than other numerical procedures. Constant strain triangles are cheaper to generate than frame elements, and although it may take a large number of these elements to model a region with a high strain gradient, they are mathematically very stable. Generally, civil engineers in their applications will prefer not to deal with high stress concentrations in their numerical analyses, as, for example, a point load applied to a concrete pier, since the design will call for a suitable spreading of this load by a steel bearing plate. A constant strain triangle model will provide these plate bearing stresses for design. In the rugged design world of the civil engineer, where structural loadings pertain to a natural environment, and where geometries are created with concrete formwork and steel bolts, lower order elements for bending, constant stress and strain, and three-dimensional stress analysis are the appropriate design tools.

5.3 Programming language

ANSI FORTRAN 77 was selected as the programming language for the three programs developed. It is the current standard of FORTRAN and should be used in any software development done in civil engineering. FORTRAN 77 is supported by the latest mainframe computers, and recently compilers have been developed for mini and microcomputers as well [2]. In the past, mini and microcomputers used FORTRAN 66 with various extensions which appeared to have the capabilities of FORTRAN 77 [39], but there was no standard and programs written on such a system were not necessarily transportable.

Another reason why FORTRAN 77 was selected was because it easily handles numerical calculations. Complex algebraic expressions can be written on a single line, and many intrinsic functions can be used to assist in the calculation or testing of data. Since FORTRAN 77 is a compiled language (unlike BASIC or LISP, which are interpreted), execution is fast. A compiled language is one where the source code, written in a high-level programming language, is run through a compiler, which translates it into low-level binary code before the program is executed. The binary code then exists as a complete file and is executed as a single unit (run unit). However, interpreted languages are written in high-level code and are translated to machine language, one line at a time, during execution. No object file or run unit is created, and the program has to be interpreted

each time that it is executed, whereas a compiled program need be compiled only once. For this reason, compiled languages are executed more quickly than interpreted languages.

There are other reasons why such a strong emphasis is placed on using FORTRAN as compared to other computer languages such as PASCAL, ASSEMBLER, LISP, C, etc. [22]. Programs used for engineering applications are very large, and are therefore built up of small subroutines. Such structured programs are easy to write and debug as each module is written and tested separately and then incorporated into the complete program only when it is free of bugs. Computer usage for engineering application is CPU intensive (i.e. the computer is used mainly for performing extensive numerical calculations rather than for input and output or the manipulation of data) and billions of dollars per year are involved. FORTRAN is the most efficient language for such applications.

Engineering programs generally have large output files and the programmer requires complete control over the formatting so that this output can be presented in an intelligible manner. Design engineers use the information produced by the application programs directly, and this material is usually included directly in an engineering or insurance report. Formatting in FORTRAN is very easy and this is another reason why FORTRAN is so popular.

Engineers also use very large input data files which contain a large number of alphanumerics on "header cards". Therefore, the programming language must have the ability to perform internal input/output (i.e. to preread data statements), and to easily manipulate character strings. Typical finite element problems require data files exceeding 5,000 lines of input, so that the input specifications must be well designed so that they are easy to use and so that they detect any input errors before executing the program. This can only be accomplished by internal I/O and character string manipulation supported by FORTRAN.

Analysis and design software almost always involves the extensive use of bulk storage. The program must be able to create and store large quantities of intermediate data during the execution of an application program. Provisions in the program must exist to open and close files, several of which are "scratch files" never seen by the user. Most languages have provisions for opening and closing files, but FORTRAN is probably the most convenient to use since it allows the programmer to specify whether the file to be used is new, old, scratch, or unknown, whether the access is sequential or direct, and the action which should be carried out if an error condition exists. Many programs must be capable of stopping, for later restart, with all files intact, and FORTRAN is the most convenient language which meets all of these requirements.

Several high-level languages such as BASIC, LISP, PASCAL, and C were considered for writing the computer programs for this thesis, but most were rejected because they did not satisfy the requirements outlined above.

5.4 Logistics

The pre- and postprocessor was initially developed with the intention of using the programs on the mini or microcomputer within the design office and electronically transmitting the input and output data files to and from a service bureau mainframe, where the numerical analysis would be carried out. The Honeywell CP6 mainframe at Carleton University was used to simulate the "office computer" while the CYBER mainframe computers at CDC were used to perform the STARDYNE FEM analysis, and hence represented the service bureau computer. The CP6 and CDC computers could not communicate directly, and hence the communication was accomplished with an OSBORNE 1B microcomputer using the CROSSTALK [33] communications software, an acoustic coupler, and a Bell Telephone line to transmit data at 300 baud. The configuration is shown in Fig. 5-2. Under actual working conditions, the office computer would communicate directly with the service bureau mainframe as shown in Fig. 5-3.

All program source code was written on the OSBORNE using the WordStar word processing program [32] and was then uploaded (transferred from the microcomputer to the host system) to Carleton's CP6 system, where it was debugged and tested. The OSBORNE was then used as a remote terminal to run the pre- and postprocessors, and to transfer data between the two mainframe computers. Since the OSBORNE does not support graphics, the program MOVIE was run directly at Carleton University using the Tektronics or the VGT100 graphics terminal.

The micro and mainframe computers were also used as word processors for all reports. WordStar was used to prepare the text files and the rough drafts were printed on a letter quality printer. The final draft was transferred to CP6 where T_EX word processing [29] and the Laser printer were used.

Figure 5-2: Actual configuration of computers and how they were used.

Figure 5-3: Ideal configuration of computers requiring access to service bureau software.

Chapter 6

General programming features

FORTRAN 77 has several useful features which greatly simplified the development of the three computer programs discussed in this thesis. The description of how these programs work and how they are used is given in the following chapter. This chapter discusses some general programming concepts of FORTRAN 77, and the examples presented were not necessarily used in the programs developed by the author.

6.1 OPEN and CLOSE statements

Most interactive computer programs have some provision for reading input data from a file and writing output data to another file. This could be done externally, i.e. by issuing the appropriate job control commands before executing the computer program, but such an approach is not convenient to a program user. It is more convenient to the user to open files interactively during execution in response to clear computer instructions, or in the case of utility files, which are important to the execution but not directly meaningful to the user, to not have to consider these at all. At the same time, several tests can be performed and the program can warn the user if an error condition occurs during an attempted opening of a file so that corrective action may be taken.

A file is an external source from which data may be obtained, or an external destination to which data may be sent. However, data cannot be transferred to or from a file until that file is "connected" to the program being executed. The principal purpose of file connection is to designate a "unit number" to be used by the program in referring to that file (i.e. in READ, WRITE, REWIND, and BACKSPACE statements), and to establish certain properties that affect data transfer. A file becomes connected after the execution of an OPEN statement, which must contain the unit number, and may also contain several optional parameters. The syntax with a complete list of options and an explanation of what they are used for may be found in any FORTRAN 77 reference manual or textbook, and therefore will not be discussed here. Although most options may be omitted from the OPEN statement and appropriate default values will be

automatically set, there are several parameters which are very useful. The syntax for such an OPEN statement is as follows

```
OPEN (u, NAME=fid, STATUS=type, ACCESS=ax, FORM=ft, ERR=s)
```

where *u* is the unit number and must be specified, *fid* is the name of the file, *type* is the type of the file and may be either "OLD", "NEW", "SCRATCH", or "UNKNOWN", *ax* specifies whether the file has "SEQUENTIAL", "DIRECT", "KEYED", or "RANDOM" access, *ft* specifies whether the file is "FORMATTED" or "UNFORMATTED", and *s* gives the statement number to which control of the program jumps if an error condition occurs during the attempted execution of an OPEN statement. Thus, for example, the statement

```
OPEN(1, NAME=STRING, STATUS='OLD', ACCESS='SEQUENTIAL', FORM=
+ 'FORMATTED', ERR=100)
```

opens a formatted, sequential file having unit number 1, which has been previously created by the user. The name of this file is contained in the character variable "STRING". If this file is not found, an error condition occurs and program execution jumps to location 100, which will probably be a command to print an error message. Other files, such as new files used to store data, and scratch files for storing temporary results, may be created in a similar manner.

Unit numbers, *u*, are generally arbitrary consecutive numbers starting at 1. Several other devices on the computer system, for example, the terminal and the printer, have been assigned unit numbers by the manufacturer. The unit numbers for these devices are usually 3-digit numbers, and program developers are advised to avoid selecting numbers at random in case one of these device numbers is accidentally used. The "name" is also left to the user's discretion, and an alphanumeric variable can be used, e.g. "FILENAME", so that existing data files can be read from the user's account. The variable "FILENAME" must be declared as a character variable, for example by the declaration statement

```
CHARACTER FILENAME*36
```

and provision must be made within the program to set the value to the name of the file which exists in the account, or which is to be created when the program is executed. The old files which are accessed by the program during any execution must be present in the user's account. Such files are typically created manually by using a text editor and typing in all the relevant data, or automatically as the output of some other program.

To disconnect a file from the program being executed, the CLOSE statement must be used. If this statement is omitted and the program terminates execution in a normal manner, any file which has been opened will be closed by default when the program execution stops. This means that OLD or NEW files will be saved, whereas SCRATCH files will be deleted. The correct syntax is

```
CLOSE (UNIT=u, STATUS=dis, ERR=s)
```

where *u* is the unit number of the file to be disconnected, *dis* is either "DELETE" or "KEEP", and *s* is the statement number to which execution jumps if an error occurs during the attempted closure.

If a program execution terminates abnormally, it is possible that the user's computer account will be left with partially completed scratch files, which would have been deleted had the program executed properly. This problem is difficult to avoid completely, however it is important that the next execution of the program not be halted because a "NEW" file which is being opened already exists. It is good practice for the programmer to initially open all scratch files (and the output file) as "status = unknown", then delete the file, and open as "status = new". An example of such code is as follows.

```
OPEN(UNIT=6, NAME='IFILE', STATUS='UNKNOWN', FORM='FORMATTED')
CLOSE(UNIT=6, STATUS='DELETE')
OPEN(UNIT=6, NAME='IFILE', STATUS='NEW', FORM='FORMATTED')
```

Alternatively, a dialogue may be used which informs the user about the status of any files which are to be created as "NEW" but which already exist, or files which are to be used as "OLD" but which have not yet been created. In the programs developed by the author, several files are connected and disconnected during the execution of each program. In order to reduce the redundancy of the FORTRAN code, to prevent any possible errors which the user might make, and to make the task as simple for the user as possible, subroutine OCHECK was developed. A listing is given in Fig. 6-1.

```

C **** OCHECK **** P. KALAB, VERSION 1.2; SEPTEMBER 19, 1983. ****
C SUBROUTINE WHICH OPENS FILES AND CHECKS FOR ERROR CONDITIONS. THE
C USER HAS 3 CHANCES TO CORRECT ANY MISTAKES HE MAY HAVE MADE.
C * SYNTAX * OCHECK(I1,CHAR1*36,CHAR2*7,CHAR3*7,I2)
C * VARIABLES * I1 = DEVICE NUMBER; CHAR1 = FILE NAME; CHAR2 = TYPE;
C CHAR3 = ACCESS; I2 = ERROR CODE (0=NONE, 1=ERROR)
C
SUBROUTINE OCHECK(DEV,STRING,TYPE,AX,ER)
IMPLICIT REAL(A-H,O-Z)
INTEGER DEV,ER
CHARACTER STRING*36,TYPE*7,AX*10,RESP*1
IREP=0
10 CONTINUE
OPEN(UNIT=DEV,NAME=STRING,STATUS=TYPE,ACCESS=AX,FORM='FORMATTED',
+ ERR=100)
ER=0
RETURN
100 CONTINUE
IF(TYPE(1:3).EQ.'OLD') THEN
PRINT *,
+ 'ERROR - File is not in directory or has incorrect access.'
ELSE IF(TYPE(1:3).EQ.'NEW') THEN
PRINT *, 'ERROR - File already exists in directory.'
○ PRINT *, 'Do you wish to overwrite it?'
READ '(A1)',RESP
IF(RESP.EQ.'Y') THEN
OPEN(UNIT=DEV,NAME=STRING,STATUS='OLD',FORM='FORMATTED')
CLOSE(UNIT=DEV,STATUS='DELETE')
OPEN(UNIT=DEV,NAME=STRING,STATUS=TYPE,ACCESS=AX,
+ FORM='FORMATTED',ERR=100)
RETURN
ELSE
END IF
ELSE
PRINT *, 'ERROR - Unable to open file.'
END IF
IREP=IREP+1
IF(IREP.GT.3) THEN
PRINT *, 'Sorry, unable to open file specified.'
ER=1
RETURN
ELSE
PRINT *, 'Enter correct file name:'
READ '(A36)',STRING
GOTO 10
END IF
END

```

Figure 6-1: Source code for subroutine OCHECK, which opens all files.

6.2 Internal input/output and character manipulation

One important feature of FORTRAN 77 is the ability to read the same line of input data by using different formats. This is useful in extracting keywords from input data, which then determine subsequent flow of the program, or in detecting incorrect input, and thereby preventing run-time errors from occurring.

The general procedure is to read a line of input as a character string into a character variable, and then use that variable as a "unit number" for re-reading the string using the proper numerical format. For example, if the coordinates of a series of nodes are to be read from a file written in a format where the word "NODE" is followed by the node number and the X, Y, and Z coordinates, such as

```
NODE  23   15.63   12.91   -2.77
```

the program code used to read the appropriate values might be.

```
      READ(1,10) STRING
10    FORMAT(A72)
      IF(STRING(1:4).EQ.'NODE') THEN
        READ(STRING,20) I,X(I),Y(I),Z(I)
20    FORMAT(4X,4G)
```

Closely associated with the internal read is the ability to handle characters, i.e. to selectively extract individual characters from a character string. This concept has been shown in the above example, where columns 1 to 4 of the variable STRING are tested to determine whether they equal the string "NODE". Another example is a segment of code which tests a line of input to determine whether the user wishes to analyse a planar truss, planar frame, space truss, or space frame, by testing for the appropriate first two letters of a line of input as follows:

```
      READ(1,10) LINE
10    FORMAT(A72)
      IF(LINE(1:2).EQ.'PT') THEN
        GOTO 100
      ELSE IF(LINE(1:2).EQ.'PF') THEN
        GOTO 200
      ELSE IF(LINE(1:2).EQ.'ST') THEN
        GOTO 300
      ELSE IF(LINE(1:2).EQ.'SF') THEN
        GOTO 400
      ELSE
        END IF
```

Once a match is made, control jumps to the appropriate location in the program, otherwise the statements following the IF block are executed.

6.3 Branches and loops

One important improvement over previous versions of FORTRAN is the ability to perform multiple tests using the IF...THEN...ELSE structure, as may be seen in the preceding example. The older versions of FORTRAN only permitted a single IF test, and the multiple tests had to be performed by using several IF and GOTO statements, which made the code difficult to read and debug. Since both methods are acceptable, the one that is used depends on the programmer's personal programming style.

When a number of statements must be executed repeatedly, it may be convenient to use a loop structure. The most common is the DO loop, which repeatedly executes a block of statements between the DO statement and the loop terminus. The number of repetitions depends on the values of the control variables in the DO statement. DO loops are convenient if the number of times that the loop must be executed is known or can be determined. However, when the number of iterations is unknown, a more complex loop and test must be developed. For example, the following segment of code must scan a data file until it finds the word "DEFLECTIONS" in the appropriate location, and if it is not found, control of the program jumps to location 9000, which will print an appropriate error message.

```
450  CONTINUE
      READ(1,132,END=9000) LINE
      IF(LINE(57:77).NE.'DEFLECTIONS') GOTO 450
```

6.4 Miscellaneous commands

There are many other commands available to the programmer. The BACKSPACE, for example, positions the pointer in a sequential file at a record preceding the one which has just been read, while the REWIND command positions the pointer at the very beginning of the file. In this way, sequential files may be read as many times as required.

The disadvantage of using sequential files is that records can only be read one after another, so that if information from the middle of the file is required, a large number of unneeded records must be read first. Under such circumstances, it is better to use direct access files, which are read or written according to the key number of each record. The records may thus be accessed in any order and as many times as required, and the files are used in the same way as arrays,

with the advantage that the size remains flexible and expands automatically to accomodate more information as it is being entered. Direct access files are usually used for intermediate results, and are therefore usually "SCRATCH" files. Input and output data files which are saved would rarely be used as direct access, and are almost always defined as sequential access files

There are many other useful features of FORTRAN 77 which the programmer has at his disposal [24]. The examples presented in this chapter illustrated only some of the more important concepts, and were intended as a guide to show how programs may be written in order to make use of the full capabilities of FORTRAN 77.

Chapter 7

Program description and use

The coupled shear wall system as shown in Fig. 7-1 was used to illustrate how the preprocessor, postprocessor, and translation programs are run. The example does not represent a real structure, but the dimensions were selected after considering several shear walls described in various references [3,11,38,45]. The geometry of the sample problem is typical for a shear wall system which could be used in a medium-rise apartment building or an office building. The mesh generated is coarse in order to minimize the computer time and resources required for the analysis, and to keep the time required for the transfer of data as short as possible. If this wall was analysed in a design office, the mesh would be finer.

The load magnitudes are also typical for this type of structure [45]. The applied loads depend on the geometry of the entire building, the location of the wall within that building, the location of the building itself, the foundation conditions, and several other factors. More extensive preprocessing by experts in the field of shear wall design could conceivably further automate the design of such systems.

Figure 7-1: Diagram of coupled shear wall system modelled by the preprocessor.

Figure 7-2: Flowchart of the preprocessor operating in the normal data-entry mode.

GEOMETRY DATA

CONCRETE PARAMETERS:

1) Density = 2400. kg/cu.m; 2) Strength = 30.0 MPa; 3) E = 27691. MPa

WALL WIDTHS:

4) Wall 1 = 6.000 m; 5) Gap = 2.000 m; 6) Wall 2 = 5.100 m

7) Number of storeys = 6

8) No. of elements in horizontal direction in Wall 1 = 4

9) Maximum permissible aspect ratio = 10.0

STOREY	WALL THICKNESS (m)	STOREY HEIGHT (m)	BEAM HEIGHT (m)
1	.200	4.000	1.500
2	.200	3.000	.500
3	.200	3.000	.500
4	.200	3.000	.500
5	.200	3.000	.500
6	.200	3.500	1.000

APPLIED LOADS:

STOREY	VERTICAL UDL (kN/m)	LATERAL CONC. LOAD (kN)
1	70.000	80.000
2	70.000	150.000
3	70.000	180.000
4	70.000	240.000
5	70.000	340.000
6	70.000	380.000

LINEARLY VARYING LATERAL LOAD (kN/m):

7) Bottom = .000 ; 8) Top = .000

Highest aspect ratio = 6.8

Max. allowable ratio = 10.0

No. of elements in horizontal direction in Wall 1 = 4

No. of elements in horizontal direction in Wall 2 = 3

No. of elements in vertical direction per storey = 4

No. of nodes: 225

No. of elastic beams: 6

No. of rigid beams: 24

No. of quad. plates: 168

Figure 7-3: Summary of input parameters displayed by the preprocessor.

7.1 Preprocessor

The preprocessor is used to facilitate the preparation of input data for analysing concrete coupled shear wall systems with the STARDYNE finite element program. The present dimensioning of the program enables the user to analyse structures which have a maximum of 50 storeys, however the program was designed so that a minimum number of changes would be required to modify the source code to process larger structures.

All input is in the S.I. units of kilonewtons, metres, and megapascals. The dialogue between the computer and the user was designed so that it is brief yet complete. The number of parameters which the user must input has been kept to a minimum, so that in a typical run the user must respond to less than 20 prompts of which half are used for the input of data and the remainder are used to control the flow of the program.

The preprocessor can operate in two modes. In the normal mode, all prompts are printed and data are entered in a specific sequence, as shown in the flowchart in Fig. 7-2. The second mode, or edit mode, enables the user to edit the geometry or load data by entering "GEOM" or "LOAD", respectively, when a numerical answer is expected. This "override" is useful when the user wishes to bypass the prompts. A summary of the input parameters as displayed by the two edit modes is shown in Fig. 7-3. The complete program listing of the preprocessor source code is shown in Appendix A, and the dialogue between the computer and the user for the sample problem is shown in Appendix D.

Extensive use has been made of subroutines which perform specific housekeeping and input/output tasks. The program has been designed in a modular fashion so that the main program is used mainly to initialize variables and then call individual subroutines to perform the rest. The subroutines, the most important of which are listed in Fig. 7-2, make full use of the programming capabilities permitted in FORTRAN 77.

7.1.1 Initialising variables and opening files

The main program first initializes variables and prints the program name and general instructions. The program may run in two modes where all prompts are printed in the normal data-entry mode, or the prompts are suppressed and the user may enter data directly in the edit mode. There are three special commands which will override the normal program flow. "STOP" (or "S" when used after a "Y/N" prompt) will immediately abort the run, and this command may be entered at any time. For this reason, the user cannot use file names which begin with "STOP". The second special command is "GEOM", which enables the user to switch to the edit mode of the geometry

data. This command may be entered at any time when a numerical answer is expected except when loads are being processed. All geometry prompts may be bypassed with this command and data may be entered directly from the edit mode. The third command is "LOAD", and it is analogous to the GEOM command except that it is used to activate the edit mode for load data. It may be invoked at any time that numerical input is expected, even before the geometry has been specified, but the geometry data then has to be entered using the edit mode.

The first prompt requests the user to specify whether he is carrying out a "Geometry only" run. If the answer is affirmative, only the geometry data will be prepared and the loads will not be generated. To process complete structures with loads, the user must answer with a negative. The next one or two prompts, depending on whether the loads are to be processed, ask the user to enter the file names where the data will be stored. The STARDYNE input file contains the geometry, loads, and control parameters for the FEM analysis, and it is this file which is transmitted to the computer system where STARDYNE operates. The second file is the postprocessor command file, which is not generated if no loads are processed. This file contains data needed by the postprocessor. The user must use legal file names, otherwise a run-time error will occur. The file names must also be less than 36 characters in length, otherwise the name will be truncated.

The opening of files is done by subroutine OCHECK (open files and check for errors). The arguments for the subroutine are the device number of the file to be opened (unit 1 for the STARDYNE input data and unit 2 for the postprocessor commands), the name of that file, the type (new or old), the access (sequential or direct), and an error indicator. If a new file is to be opened and a file of that name already exists in the user's account, the user is advised of this fact and is given the option of overwriting the existing file or opening a file with a different name. If an old file is to be opened and does not exist in the user's account, an appropriate error message is printed. If the file of a desired name cannot be opened, the user is given three chances to enter a correct name which has a maximum of 36 characters. After successfully opening the file, the error indicator is set to zero, control returns to the main program, and normal execution proceeds. However, if the file could not be opened even after three tries, the error indicator is set to 1 and execution is stopped by the main program. Unfortunately there are some illegal names which cannot be used for files (such as "LP") because it will cause a run-time error to occur. No provisions have been made in the program to filter these out, and the user is advised to use "sensible" names. This subroutine was found to be necessary, because frequently a user makes a typing error when typing, and if an ordinary OPEN statement was used, various errors could occur and files could accidentally be deleted. Subroutine OCHECK therefore provides some error-checking and permits limited recovery should errors occur.

7.1.2 Input of data

The next series of prompts asks the user to input the numerical geometrical properties of the shear wall system. The subroutine which handles the input and displays the prompts expects a certain number of numeric data fields in each input line. The numerical input is written in free format where fields of integers, real numbers, or exponential numbers, are separated by spaces or commas. If fewer fields than expected are entered, or if characters other than the commands mentioned above are used, the program will print an error message, the erroneous input line, repeat the prompt, and ask the user to re-enter the correct data. If a blank line is entered, all parameters are set to zero. In some cases, blank lines are used to terminate the input of large blocks of data, but the user is always advised of this.

The input and output is accomplished by subroutine Q (question). The first argument passed to Q is a character string with a maximum of 72 characters, which is the prompt to be printed. The second argument is an integer constant which tells the subroutine how many numeric arguments to expect as input, and the third argument is a character variable which reads the input from the user as a character string of at most 72 characters. Subroutine Q in turn calls other subroutines which read the input from the terminal, scan the input for any errors, and print warning or error messages as appropriate.

The first numerical prompt asks for the material density in kg/m^3 and the strength in MPa. Young's modulus is calculated automatically, and all units are converted to the consistent units of kN and metres before the data are processed by STARDYNE. All geometry data must be greater than zero.

The next prompt asks for the width of Wall 1 (on which the lateral loads act), the separation between the walls, and the width of Wall 2 in metres. Wall 1 must be as wide as or wider than Wall 2, otherwise the dimensions for the two walls will be interchanged automatically.

The third prompt requests the user to enter the number of floors in the building. The limits are 1 to 50 inclusive with the present configuration of the program, and if these limits are exceeded, an error message is printed and the user is asked to re-enter correct data before proceeding any further.

The user is then asked to provide the out-of-plane thickness of the shear wall system, the storey height, and the beam height. The storey height is the distance between the base of one floor and the base of the floor above. The beam height is the distance between the top of the opening separating the two walls and the base of the floor above. The beam height must be less than the floor height, so that the system cannot be a solid block without openings.

The next question asks whether the floors above the first floor have a constant floor height, beam height, and wall thickness. A "Y/N" response is required here. If the answer is affirmative, the user is asked to input all parameters as for the first floor. If the answer is negative, the user is asked to enter each floor where the geometry changes and the new parameters. It is not necessary to input data for all floors, because those floors not entered will have the same geometry as the storey below which has last been defined. Any number of data lines may be entered in any order as new values for a particular floor will overwrite values entered previously. The floor number must be within the range specified, all values must be greater than zero, and the beam height must be less than the storey height. If any of these conditions are not satisfied, an error message is printed and the user is asked to enter correct data. Entry of data is terminated with a blank line.

The geometry data are input in this manner for the following reason. Most buildings have a uniform geometry for all floors except the first, which may sometimes have more headroom. If the geometry is constant, the user only has to enter one set of data, and by answering the prompts which tell the computer that the geometry for all other floors is constant, and that it is the same as for the first floor, the computer fills in the rest. Alternatively, the storeys above the first floor may have a constant geometry which differs from that of the first floor, in which case another set of data must be entered, and the computer again fills in the rest. If, however, the geometry changes at several floors, as with a penthouse, mechanical floor, or basements, the user only needs to enter the data for the floor where the geometry changes, and the computer automatically fills in the data for the remaining floors. Hence the program is very versatile since it handles any geometry, and is very easy to use since minimum input data is required. A similar approach has been used when entering the data for the vertical UDL loads.

7.1.3 Editing of geometry data

Once the geometry has been entered, it is displayed and the user has the option of making changes. Editing takes place in two stages. First, the user must specify that he wishes to invoke the editing mode after the appropriate prompt, and then may change the values for code numbers 1 to 9 (which precede the parameter to be altered and are followed by a right parenthesis) by entering the code number and the new value. The new value overwrites the old value and this process may be repeated as many times as required. Editing of this section ends when a blank line is entered. The user then has the option of changing the wall thickness, storey height, and beam height by entering the storey number and all three new parameters. Unfortunately this may be tedious if extensive changes are required, but more complicated source code or a different display would be required to make editing of this section easier. Editing ends when a blank line is entered.

Display of data and editing is performed by subroutines PG (print geometry) and PL (print loads) for the geometry and load data, respectively. The subroutines first test several variables to determine where they have been called from and whether the call is legal. Error messages are printed if the subroutine cannot be invoked at that particular time, or prompts are printed which lead the user through the editing procedures. The subroutines display the data and allow the user to make changes or enter data directly from this level if the prompts have been bypassed. This option is useful if several similar walls are analysed and only some of the parameters need to be changed. The parameters keep the values from the preceding run of the same session. Changes to the data are made by entering a code number and the new data. An IF...THEN...ELSE structure is used to modify the specified parameters and test whether they still fall within the specified limits. An error message is printed if the code number is out of range.

Three variables appear in the display of the geometry data which have not been entered by the user, but which may be altered if the user so wishes. These are the Young's modulus of elasticity, E , the number of elements in the horizontal direction in Wall 1, and the maximum allowable aspect ratio. E is calculated automatically from the parameters DENSITY and STRENGTH by the equation [7]

$$E = \text{DENSITY}^{1.5} 0.043 \sqrt{\text{STRENGTH}}$$

but may be altered if the user so wishes. The number of elements in the horizontal direction (NEH) and the maximum aspect ratio (AR) are used by subroutine ASPECT to calculate the coarseness of the finite element mesh. By default, NEH is set to 4 and AR, which is the ratio of the larger dimension of the finite element to the smaller dimension, is set to 10.0. A larger number of elements or a smaller aspect ratio or both results in a finer mesh and hence longer computing time and more accurate and costly analysis. The converse is also true. The limits are one element and an aspect ratio of 1.0, but such a fine mesh should never be specified because it will be nearly impossible to generate with the iteration limits set by the program (200 cycles maximum for each of eight critical elements).

7.1.4 Error checking and input of load data

Once the editing of geometry data has been completed, the main program checks the data which have been entered so that none of the restrictions, which have already been discussed, are violated. If any violations are encountered, the user is asked to correct the data. Also certain limits have been set on the storey height and beam height to prevent the user from entering unreasonable geometry parameters. These limits are a maximum to minimum storey height not

exceeding 5, maximum to minimum beam height not exceeding 10, and maximum wall height to minimum beam height not exceeding 35. These ratio limits are arbitrary, and if they are exceeded, a warning message is printed advising the user of the fact. The user is, however, given the option to proceed on the understanding that the mesh may be impossible to generate within the number of iterations permitted by the program, or the mesh may be so fine that it exceeds the computer limits, either memory for the data files, or the STARDYNE node, beam, and quadrilateral plate limits.

Once the geometry has been correctly entered and edited, the user inputs the loads. This step is bypassed if the user specified a "Geometry only" run. All lateral loads act on Wall 1 and vertical loads act on each storey level such that toward the wall and down are considered positive directions. The input is modified for STARDYNE, which considers up and to the right as positive directions. Vertical UDL is constant over one floor, but a unique load may be applied to each floor. As with the geometrical properties, the user need enter data only for floors where the loads change, and the other loads are generated automatically.

The second prompt asks for the linearly varying lateral load at the base and at the top. This load configuration may be useful when wind pressure is considered, however the limitation is that this is a trapezoidal load applied to the entire height of the building.

The third prompt asks the user to input concentrated lateral loads acting at each storey. Since each load is unique, the user must enter the floor and load magnitude for all floors where such a load is acting as no loads are generated automatically. The data may be entered in any order and corrections may be made by re-entering a particular storey and load. Entry of data ends with a blank line.

7.1.5 Checking the aspect ratio

The user is then given the option of editing the load data in a similar manner as the geometry was edited. He then has one more chance to edit the geometry data. The program then generates the mesh parameters and prints a summary. This is done by subroutine ASPECT. The arguments passed to it are NEH, AR, the geometrical properties of the structure, and the maximum and minimum dimensions of the floor height and beam height. The subroutine then examines the critical elements which were established from NEH and the maximum and minimum wall dimensions. For this reason, Wall 1 must be as wide as or wider than Wall 2. The aspect ratio of the critical elements is compared to the maximum permissible aspect ratio, and if this limit is exceeded, the mesh is made more fine. Eight critical elements are checked and one hundred iterations are permitted in setting the dimensions for each critical element. Since the calculations for one element may affect

the aspect ratio of elements already checked, a second pass is made over all elements to ensure that the aspect ratio is within the required limits. If the limits are exceeded, warning messages are printed telling the user where the problem was encountered and the user is given the option whether to proceed or alter the geometry data. Although provisions have been made to discourage the user from entering absurd data (e.g. a floor height of 10 m with a 1 cm beam height), it is nevertheless possible to specify parameters which will exceed the CP6 disk space limits thereby causing a run-time error to occur, or which will exceed the limits of STARDYNE should the analysis be performed. No provisions have been made to prevent such abuse as it was assumed that the user would exercise some engineering judgement.

7.1.6 Calculation of geometry

Once subroutine ASPECT is finished, control is passed to subroutine GEOM, which prepares the geometry data according to the STARDYNE reference manual. Many input parameters are generated automatically, and if the user should, for any reason, wish to change these values, the changes would have to be made by editing the input data file, or by changing the program source code. The start, description, and material properties cards are written, and the nodal coordinate table is generated. This is followed by the restrained boundary, elastic beam, rigid beam, and quadrilateral plate connectivity tables. The numbering of the nodes and elements is shown in Fig. 7-4.

The model assumes that the wall is fully restrained at the base and that there is no out-of-plane movement anywhere in the wall. The two walls are connected by elastic beam elements so that the end forces and moments may be directly calculated by STARDYNE. The wall adjacent to the beam is divided into two elements in the vertical direction. To ensure that the ends of the beam rotate in one plane and not as two separate elements lying between three nodes, rigid beams have been used as shown in Fig. 7-5. A summary of the number of nodes, beams, and quadrilateral elements is printed, and the user again has a choice of interrupting the program and editing the geometry data. This provision is useful if the user finds that the mesh is too fine or too coarse (from the number of nodes and elements).

Figure 7-4: Nodes and elements as they are numbered by the preprocessor.

Rigid links A-B-C and X-Y-Z ensure that the ends of the beam lie in a single plane. The elastic beam connecting the two walls is attached to nodes B and Y, but the physical dimensions of the beam simulate an element occupying the space ACZX. The walls are divided into 2 beam elements, BE, per elastic beam, and an arbitrary number of wall elements, WE, in the vertical direction. Vertical UDL loads are applied to the nodal row C-Z, which is the dividing line between the floors of the building.

Figure 7-5: Simulation of the beam-wall connection.

Figure 7-6: Flowchart of the postprocessor.

7.1.7 Calculation of loads

Subroutine LOAD is then used to calculate the loads acting on the structure. It first opens two direct access scratch files where intermediate nodal loads are stored (unit 3 for vertical and unit 4 for horizontal loads). These files are used in the same way as arrays such that the line numbers correspond to the nodes at which the loads act. The gravity loads, based on the density and the geometrical properties of the mesh, and the vertical and horizontal applied loads are placed on the appropriate nodes on the scratch files as they are calculated. The scratch direct access files have been used because they permit the data to be handled in the same way as arrays, yet they are not fixed in size but may vary with the number of the nodes.

Subroutine LOAD is executed only if the user specified that he did not wish a "Geometry only" run. After calculating the loads, it then confirms the file names where the data are stored, and asks the user whether he wishes to process a new case. If the answer is negative, execution stops. If the answer is affirmative, the program loops back to the start. However, the variables are not re-initialized, so that they have the values entered in the preceding run. In this way, the user may make minor changes in the edit mode bypassing all prompts, and it is possible to process several structures of similar geometry and loads with minimum effort.

7.2 Postprocessor

Before the postprocessor may be used, the input data must be transferred to the computer system running the STARDYNE application program, the STARDYNE analysis must be performed, and the raw output data file or the important tables must be transferred back to the computer system where the postprocessor is running. Even with the small example used here, the time required for the transfer of data is long. At 300 baud, the input data file, shown in Appendix E, required 18 minutes for one transfer, and selected portions of the output data file, shown in Appendix F, required $1\frac{3}{4}$ hours for one transfer. Since the data had to be transferred first from one mainframe to a microcomputer and then from the microcomputer to the other mainframe, the total transfer time was doubled. If the entire STARDYNE output file was transmitted at 300 baud, the total transmission time, both ways, would require approximately 24 hours for this problem. As the number of nodes and elements increases, the transmission time required increases exponentially. Most of the data in the output file are not required by the postprocessor, and hence it need not be transmitted. The complete file should, however, be printed and used by the engineers in the design office if other information than what is presented by the postprocessor is required. It should be remembered that the units are in kN and metres. The data which are required by the

postprocessor are the "Static deflections", "Beam element loads", and "Quadrilateral stresses and forces" tables

When running the postprocessor, the user must first enter the file names where the STARDYNE output data and the postprocessor commands are stored. Data are then read from these files. If any errors are encountered, which may have been introduced during transmission or because incorrect files have been opened, the user is notified and execution stops.

The postprocessor is designed along similar lines as the preprocessor, and uses some of the same subroutines such as OCHECK, Q, CN72, and WL. The flowchart of the postprocessor is given in Fig. 7-6 and the source code is given in Appendix B.

The postprocessor first uses subroutine OCHECK to open the file where the STARDYNE output is stored (unit 1) and the file where the postprocessor control commands are stored (unit 2). The control file includes the geometry and load data as listed in the edit mode by the preprocessor, and a section called "raw data" which lists some of the same parameters in a different format. The reason for this duplication of data is that even if some minor changes were made in the output format of the preprocessor, the source code of the postprocessor would have to be altered considerably, since the data would no longer be present in the expected fields. By having a special "raw data" section which is read in free format, the postprocessor becomes independent of the text format of the preprocessor.

The STARDYNE output file is then read and the information is stored in various arrays and a direct access scratch file, which is used to store the centroid of each element. The stress data are converted from kN/m^2 to N/mm^2 (MPa). If any data are missing or have been altered due to faulty transmission, the program tries to bypass them and print a warning message. If significant portions of the data are missing (i.e. entire tables), or if the data cannot be read (if the wrong files have been opened), an appropriate error message is printed and the run is aborted.

Providing that all data have been successfully read, the user is given the option of having all data printed back, or to browse through only selected sections. The data are printed in a more intelligible format to the terminal and to the postprocessor command file, so that the user can obtain a hard copy by printing that file (Appendix H). The format of the processed output is shown in Fig. 7-7, and the unprocessed output is shown in Fig. 7-8 for comparison. The lateral drift, storey geometry, applied loads, and beam forces are always printed. However, the printout of the subsequent table can be controlled by the user. It contains the listing of the elements, the centroids of the elements as measured from the toe of each wall (see Fig. 7-9), and the stress in the X direction (horizontal), stress in the Y direction (vertical), and the shear stress at the centroid (XY). Depending on the parameters which were entered by the user, this table may list

all elements, or only one element per wall. The location of the centroids has been included in the printout, because even though the postprocessor does not support graphics which could directly display the stress plots, the user may manually graph the plots from the data printed by the postprocessor. The sign convention for the forces and stresses in the elements is given in Fig. 7-10.

After all tables have been printed, the user has the option of having the data printed at different increments, i.e. view other regions of the structure. This permits the user to browse through the data which is displayed on the terminal in a concise manner rather than flip back and forth in the computer listing. Once the user finishes with a particular output file, the program gives him the option of processing a different data file. A sample output of the postprocessor is given in Appendix G.

Should the dimensions of the preprocessor be changed so that larger problems may be analyzed, the dimensions of the postprocessor and the variable MAXF must correspond to the changes made in the preprocessor. The array SL (side load) is dimensioned to the number of floors + 1 because it is used to store the lateral load acting at the base and top of each floor.

7.3 Translation program

The listing of this program is given in Appendix C. The program is used to read the input data file for STARDYNE and translate the coordinate and element connectivity data into a format which may be read by the graphics program MOVIE. Again various tests are made and error messages are printed if some data are missing. The program is very simple since it only reads from one file and writes the information to a different file without performing any calculations or extensive searches. Some of the subroutines developed for the pre- and postprocessor have also been used. The only input required from the user is the name of the file where the STARDYNE input is stored and the name of the file where the MOVIE data file will be written. The data are then translated and the user is given the option of processing a different data file.

The program MOVIE must be run on a graphics terminal such as a Tektronics, a VGT100, or a plotter. The documentation for MOVIE is extensive and therefore has not been included here. However, the main commands are DRAW or VIEW which display the data, SHRINK which shows the individual elements making up the shear wall system, and DIST which moves the origin toward or from the point of view of the user and hence changes the apparent size of the display. Other commands may be listed by using the HELP command.

 ***** STOREY NO 1 *****

Storey drift = .975 mm

GEOMETRY:

Wall 1 width = 6.000 m Wall 2 width = 5.000 m
 Beam length = 2.000 m Beam height = 1.500 m
 Storey height = 4.000 m Thickness = .200 m

LOADS:

Lateral top LDL = .000 kN/m Bot. LDL = .000 kN/m
 Conc. lat. load = 80.000 kN Vert. UDL = 70.000 kN/m

FORCES IN CONNECTING BEAM

	A+ Wall 1	A+ midspan	A+ Wall 2
Axial (kN)	187.767	187.767	-187.767
Shear (kN)	-8.630	78.630	148.630
Moment (kN*m)	-55.313	11.683	-101.947

MAXIMUM STRESS SUMMARY (in MPa):

X	Y	XY	ELEMENT
-.8819E+00			4
	-.8819E+01		4
		.1140E+01	7

WALL 1:

ELEMENT	LOCATION		STRESSES		
	X (m)	Y (m)	X (MPa)	Y (MPa)	XY (MPa)
1	.750	.625	.2511E+00	.3187E+01	.5089E+00
2	2.250	.625	-.4069E-01	-.5916E+00	.6561E+00
3	3.750	.625	-.6495E+00	-.4069E+01	.5468E+00
4	5.250	.625	-.8167E+00	-.8819E+01	.4193E+00
5	.750	1.875	-.2681E-01	.2639E+01	.3938E+00
6	2.250	1.875	.2959E-01	-.7649E+00	.8957E+00
7	3.750	1.875	-.2100E+00	-.3632E+01	.1140E+01
8	5.250	1.875	-.1942E+00	-.8416E+01	-.2981E+00
9	.750	2.875	.6981E-02	.2174E+01	.4159E+00
10	2.250	2.875	-.6132E-01	-.8300E+00	.8224E+00
11	3.750	2.875	.4716E+00	-.4298E+01	.1128E+01
12	5.250	2.875	.3723E+00	-.3174E+01	-.2579E+00
13	.750	3.625	-.1822E+00	.1822E+01	.4568E+00
14	2.250	3.625	-.6819E-01	-.8882E+00	.8489E+00
15	3.750	3.625	.4079E+00	-.4092E+01	.4147E+00
16	5.250	3.625	.1773E+00	-.2991E+01	.8136E+00

WALL 2:

ELEMENT	LOCATION		STRESSES		
	X (m)	Y (m)	X (MPa)	Y (MPa)	XY (MPa)
97	.850	.625	.9876E-01	.1125E+01	.4986E+00
98	2.550	.625	-.4013E+00	-.3369E+01	.7735E+00
99	4.250	.625	-.6496E+00	-.7883E+01	.8770E+00
100	.850	1.875	-.3003E+00	.5222E+00	.6054E+00
101	2.550	1.875	-.2843E+00	-.3439E+01	.1044E+01
102	4.250	1.875	-.6089E-01	-.7122E+01	.4995E+00
103	.850	2.875	-.6589E+00	-.6958E+00	.4995E+00
104	2.550	2.875	-.2636E+00	-.3172E+01	.8842E+00
105	4.250	2.875	-.3708E-01	-.6634E+01	.4228E+00
106	.850	3.625	-.3485E+00	-.7930E+00	.4395E+00
107	2.550	3.625	-.2286E+00	-.3191E+01	.7056E+00
108	4.250	3.625	-.5679E-01	-.6291E+01	.3945E+00

Figure 7-7: Example of processed output.

GRAVITY, VERTICAL UDL ON FLOORS, LATERAL LDF & POINT LOADS ON WALL		ELEMENT STRESSES		AND FORCES		FOR OUTPUT VECTOR		1
QUADRILATERAL ELEMENT STRESSES		PRINCIPAL STRESSES		MAX SHEAR		ANGLE		
NODES		ELEMENT FORCES		MAX			MIN	
		STRESSES						ANGLE
0	X2.	6	FX = 5.022E+01 FY = 6.375E+02 FXY = 1.018E+02 MX = 0. MY = 0. MXY = 0.	+ZFACE SX = 2.511E+02 SY = 3.187E+03 SXY = 5.089E+02 -ZFACE SX = 2.511E+02 SY = 3.187E+03 SXY = 5.089E+02	+ZFACE 3.273E+03 1.654E+02 VM 1.554E+03 3.194E+03 -ZFACE 3.273E+03 1.654E+02 VM 1.554E+03 3.194E+03	80.440		
0	X2.	7	FX = -8.139E+00 FY = -1.183E+02 FXY = 1.312E+02 MX = 0. MY = 0. MXY = 0.	+ZFACE SX = -4.069E+01 SY = -5.916E+02 SXY = 6.561E+02 -ZFACE SX = -4.069E+01 SY = -5.916E+02 SXY = 6.561E+02	+ZFACE 3.954E+02 -1.028E+03 VM 7.115E+02 1.272E+03 -ZFACE 3.954E+02 -1.028E+03 VM 7.115E+02 1.272E+03	33.612		
0	X2.	8	FX = -1.299E+02 FY = -8.138E+02 FXY = 1.094E+02 MX = 0. MY = 0. MXY = 0.	+ZFACE SX = -6.495E+02 SY = -4.069E+03 SXY = 5.468E+02 -ZFACE SX = -6.495E+02 SY = -4.069E+03 SXY = 5.468E+02	+ZFACE -5.642E+02 -4.154E+03 VM 1.795E+03 3.903E+03 -ZFACE -5.642E+02 -4.154E+03 VM 1.795E+03 3.903E+03	8.867		
0	X2.	9	FX = -1.633E+02 FY = -1.764E+03 FXY = 8.385E+01 MX = 0. MY = 0. MXY = 0.	+ZFACE SX = -8.167E+02 SY = -8.819E+03 SXY = 4.193E+02 -ZFACE SX = -8.167E+02 SY = -8.819E+03 SXY = 4.193E+02	+ZFACE -7.948E+02 -8.841E+03 VM 4.023E+03 8.471E+03 -ZFACE -7.948E+02 -8.841E+03 VM 4.023E+03 8.471E+03	2.991		
0	X2.	11	FX = -5.361E+00 FY = 5.277E+02 FXY = 7.876E+01 MX = 0. MY = 0. MXY = 0.	+ZFACE SX = -2.681E+01 SY = 2.639E+03 SXY = 3.938E+02 -ZFACE SX = -2.681E+01 SY = 2.639E+03 SXY = 3.938E+02	+ZFACE 2.695E+03 -8.377E+01 VM 1.390E+03 2.738E+03 -ZFACE 2.695E+03 -8.377E+01 VM 1.390E+03 2.738E+03	81.769		
0	X2.	12	FX = 5.918E+00 FY = -1.530E+02 FXY = 1.791E+02 MX = 0. MY = 0. MXY = 0.	+ZFACE SX = 2.959E+01 SY = -7.649E+02 SXY = 8.937E+02 -ZFACE SX = 2.959E+01 SY = -7.649E+02 SXY = 8.937E+02	+ZFACE 6.121E+02 -1.347E+03 VM 9.798E+02 1.736E+03 -ZFACE 6.121E+02 -1.347E+03 VM 9.798E+02 1.736E+03	33.041		
0	X2.	13	FX = 5.918E+00 FY = -1.530E+02 FXY = 1.791E+02 MX = 0. MY = 0. MXY = 0.	+ZFACE SX = 2.959E+01 SY = -7.649E+02 SXY = 8.937E+02 -ZFACE SX = 2.959E+01 SY = -7.649E+02 SXY = 8.937E+02	+ZFACE 6.121E+02 -1.347E+03 VM 9.798E+02 1.736E+03 -ZFACE 6.121E+02 -1.347E+03 VM 9.798E+02 1.736E+03	33.041		

Figure 7-8: Example of unprocessed output.

Figure 7-9: Location of element centroids as defined by the postprocessor.

ELASTIC BEAM FORCES

QUADRILATERAL PLATE STRESSES

Figure 7-10: Sign convention for the element forces and stresses.

Chapter 8

Conclusions and recommendations

8.1 Conclusions

This thesis has attempted to introduce the structural engineer to computer technology as it relates to engineering analysis and design. The objectives were as follows:

1. To examine the state-of-the-art software and hardware available at the present time, consider some of the future trends of the computer industry, and discuss how computers may be used in an engineering design office.
2. To describe in detail what pre- and postprocessors are, how they may be written to take full advantage of a programming language such as FORTRAN, and how they are used.
3. To illustrate the above points with a specific example.
4. To provide a limited glossary of terms used in computer-aided design.

From the discussion of the above points, it may be concluded that the use of pre- and postprocessors running on small in-house microcomputers linked to more powerful mainframe computers at a service bureau simplifies structural analysis and design, reduces the cost of analysis, enables the engineers to study several alternatives to produce an optimal design, and makes computer use more convenient for the user. While the specific programs discussed in this study may be of limited usefulness to others, the author feels that the concept they illustrate is both practical and powerful, and may be adapted to any engineering application.

8.2 Recommendations

Research and development in the computer industry is very rapid, and any report written on the subject quickly becomes outdated. As new versions of FORTRAN emerge and new software and hardware comes on the market, a number of suggestions made in this report will no longer be valid. In addition, the interconnection between computers will become more efficient and more readily available, so that many of the difficulties experienced during the development and testing stages of the three computer programs (i.e. the slow transmission rate) will no longer apply. With

this in mind, there are ~~some~~ recommendations which may benefit engineers and programmers interested in pursuing the work initiated in this study. They are as follows:

1. A survey of the computer hardware and software available on the market should be taken in specific time intervals (i.e. every five years) to determine what new developments have been made, and to predict the future trends and verify previous predictions. This report will thus be constantly updated.
2. The development of pre- and postprocessors should be intensified. It has been shown how such programs benefit the design office, and more effort should be made to acquaint other engineers who may not be aware of the savings in time and money provided by pre- and postprocessors, with the full advantages offered by computer-aided design.
3. Standards for languages other than FORTRAN should be developed. Although FORTRAN was found to be adequate for the programming required in this study, other high-level languages may offer similar, or even better features. If proper standards existed for such languages, the programmer would then have a choice of which language to use and be certain that his programs would be transportable to other systems.

Glossary of selected terms

ASCII — The American Standard Code for Information Interchange Code that has assigned a binary number to each alphanumeric character and several non-printing characters used to control printers and communication devices. There are 128 different symbols.

Baud — A unit of signalling speed equal to the number of discrete conditions or signal events per seconds. It is approximately equal to the number of bits transmitted in one second. Common baud rates used for transmission are 110, 300, 1200, 2400, 4800 and 9600.

Bit — Short form for binary digit, which can have only two possible values 0 or 1. It is the smallest unit of data recognized by the computer. All data (letters, numerals, and symbols) handled by the computer are digitized, i.e. expressed as a combination of bits. The number of bits is also used to distinguish the size of a computer by specifying the computer wordsize. For example, an 8-bit computer handles information in 8-bit segments at one time, whereas a 32-bit machine handles 32 bits at once, and is therefore faster than an 8-bit computer in dealing with units larger than 8 bits.

Bus — A group of parallel electrical connections that carry signals between computer components or devices.

Byte — The number of bits used to represent a character. By definition it consists of 8 bits. In many systems, a byte is the smallest addressable unit, however, some systems have utilities that permit bit manipulation. This feature can be very powerful in bookkeeping operations (i.e. in solving equations) in structural analysis.

Capture — The process of writing data, which is being transmitted by one computer, to the memory of the computer which is receiving the data.

Cathode Ray Tube (CRT) — A vacuum tube that generates and guides electrons onto a fluorescent screen to produce such images as characters or graphic displays on video display terminals.

Central processing unit (CPU) — Electronic components in a computer that control the transfer of data and perform arithmetic and logical operations.

Compiler — A part of the computer operating system software that reads program instructions and translates them into a machine code that the computer can read and execute.

Crash — A failure of the system due to a software or a hardware malfunction.

Database — A large collection of organized data that are required to perform a task. A typical example is a list containing steel sections and their properties.

Download — The process of transferring data from one computer to another whereby the user issues commands through the computer transmitting the data.

Emulator — A program that allows a computer to imitate a different system, thus enabling different systems to use the same data and programs to achieve the same results, but with possibly different performance rates.

Erassable Programmable Read Only Memory (EPROM).— Memory which contains "firmware" (software which has been made into hardware) which is used to control basic housekeeping tasks around the machine. By using EPROM rather than ROM chips, the manufacturer or user can erase the original program and rewrite it. Data are erased by shining a powerful ultraviolet light through the glass windows covering the chips.

Floppy disk — A flexible, flat, circular plate permanently housed in a black paper envelope with magnetic coating that stores information. Standard sizes are 8 inches, 5 $\frac{1}{4}$ inches, and 3 $\frac{1}{2}$ inches, which are now being used on some of the newer microcomputers. They are inexpensive (only several dollars each), and may be loaded and removed quickly and easily so that several might be used in any computing operation.

Full duplex — A communication circuit that permits the transmission of a signal in two directions simultaneously.

Half duplex — A communication circuit that permits the transmission of a signal in two directions, but not at the same time.

Hard disk — A magnetic disk for storing information that is not flexible, such as a Winchester disk. It is more expensive than a floppy, but is capable of storing much more information. It cannot be loaded or removed during program execution.

Hardware — The mechanical and electronic components of a computer system.

Hard-wired — A permanent physical connection between two points in an electrical circuit or between two devices linked by a communication line. When using a personal computer, local connections are typically hard-wired, whereas all connections through a modem are switched because they use telephone lines.

K — The symbol for the quantity 2^{10} , or 1024. It is usually used when referring to memory, for example 64 K means 64 Kilobytes of memory.

Large-scale Integration (LSI) — The combining of about 1,000 to 10,000 circuits on a single chip. Typical examples of LSI circuits are memory chips, microprocessors, calculator chips, and watch chips.

Microprocessor — A single-chip central processing unit.

Modem — A contraction of the words "modulator-demodulator". It is a device for performing necessary signal transformations between terminal devices and communication lines, normally used in pairs, one at either end of the communication line.

Multi-tasking — The ability of a microprocessor to perform several tasks simultaneously. It is also known as "parallel processing".

Network — A group of computers that are connected to each other by communications lines to share information and resources.

Off-line processing — The performing of part of a data processing operation on equipment that is not directly connected to the mainframe.

On-line processing — Data processing in which all the operations are performed by equipment connected to a mainframe.

Operating system — The computer programs that run the computer in contrast to the "application" programs that carry out computations. Usually the computer cannot execute the application program without help from the operating system.

Parallel transmission — Transmission whereby more than one bit is sent at one time.

Parity — A one-extra-bit code used to detect recording or transmission errors by making the total number of "1" bits in a unit of data (including the parity bit) odd or even.

Port — A physical area for the connection of a communications line. This line can be between the CPU and anything external to it, such as a printer, another computer, a modem, or another communications line. The number of ports in a system indicates the total number of peripheral devices that can be connected to a computer and operated simultaneously.

Postprocessor — A computer program which is used to translate the output of another program into a different format.

Preprocessor — A computer program which is used to simplify the input of data for another computer program.

Random Access Memory (RAM) — Memory that can both be read and written into (i.e. altered) during normal operation. RAM is the type of memory used in most computers to store the instructions of programs currently being run.

Raster graphics — Graphics whereby individual pixels (picture elements) are turned on or off to make pictures consisting of dots. In high-resolution graphics, these dots are very small and very close together so that it is difficult to distinguish the individual dots.

Read Only Memory (ROM) — Memory containing fixed data or instructions that are permanently loaded during the manufacturing process. A computer can use the data in the ROM, but cannot change it.

RS-232 Interface — The interface between a modem and the associated data terminal, as defined by the Electronics Industries Association Standard RS-232. Terminals, printers, modems, and other peripheral devices are usually connected to "RS-232 ports".

Serial transmission — Transmission whereby data is sent one bit at a time

Simplex — A communications circuit capable of transmission in one direction only

Software — Any computer program

Upload — A process of transferring data from one computer to another whereby the user issues commands through the computer receiving the data

UNIX — An operating system originally devised for multi-user minicomputers approximately ten years ago by Bell Laboratories in America. In its present form, it is a professional programmer's tool which allows users to set up whole processes of programs set in train by a single keystroke. The output of one program can be "pipelined" into the input of another. The "shell" mechanism allows any program to run any other program as if it were the operating system. XENIX is a modified version of UNIX and is used in microcomputers.

Vector graphics — Graphics whereby an electron beam is used like a pencil and actually draws shapes on the phosphor of the CRT. This gives high-quality results, but is very slow.

Word — The greatest number of bits that a computer is capable of handling in any one operation. Usually it is subdivided into bytes.

References

- [1] ANON. 1982. Let your computer debug its own programs. *Interface Age*, **8**, pp. 38-43.
- [2] ANON. 1984. Nanobytes. *Byte*, **9**(1), p. 10.
- [3] BEAUFAIT, F. W. (editor). 1974. Tall buildings. Planning, Design, and Construction. Proceedings of the Symposium of Tall Buildings, Department of Civil Engineering, Vanderbilt University, Nashville, TN, USA, November 14-15, 1974.
- [4] BENSON, T. 1983. Data communication today. *Interface Age*, **8**(3), pp. 60-81.
- [5] BLACK, W. W. 1978. Computer based models and graphics for engineering design. Conference on Computing in Civil Engineering, Atlanta, Georgia, USA, June 26-29, 1978, pp. 685-699.
- [6] BROWN, C. J. 1978. Computers in civil engineering: an overview. Proceedings of the Conference on Computing in Civil Engineering, Atlanta, GA, USA, June 26-29, 1978, pp. 1-11.
- [7] CANADIAN STANDARDS ASSOCIATION. 1977. Code for the design of concrete structures for buildings, CAN3-A23.3-M77. Canadian Standards Association, 178 Rexdale Blvd, Rexdale, Ont., Canada, 131 p.
- [8] CARSON, J. A. 1978. Computer aided design of building structures: a Canadian context. Department of Civil Engineering, Carleton University, Ottawa, Ont., Canada, 53 p.
- [9] CHRISTIANSEN, H. N. et B. J. NAY. 1980. The MOVIE system: low cost general-purpose computer graphics software. Proceedings of the Second Conference on Computing in Civil Engineering, Baltimore, MD, June 9-13, 1980, pp. 1-9.
- [10] CIVIL ENGINEERING PROGRAM APPLICATIONS, INC. 1975. A proposal for a national institute for computers in engineering. (In: BROWN [6]).
- [11] COULL, A. et B. S. SMITH (editors). 1967. Tall Buildings. Proceedings of a Symposium on Tall Buildings with Particular Reference to Shear Wall Structures, Department of Civil Engineering, University of Southampton, April, 1966. Pergamon Press, Toronto, Ont., Canada.
- [12] CYBERNET SERVICES FEMGEN REFERENCE MANUAL. 1980. Control Data Corporation, Publications and Graphics Division, 4201 North Lexington Avenue, St. Paul, MN, USA.
- [13] CYBERNET SERVICES SOFTWARE DIRECTORY. 1982. Control Data Corporation, CYBERNET Services, HQW05H, P. O. Box 0, Minneapolis, MN, USA, 75 p.
- [14] CYBERNET SERVICES STARDYNE USER INFORMATION MANUAL, VERSION 3. 1979. STARDYNE Project Office, System Development Corporation, 2500 Colorado Avenue, Santa Monica, CA, USA.
- [15] CYBERNET SERVICES UNISTRUC II REFERENCE MANUAL. 1980. Control Data Corporation, Publications and Graphics Division, 4201 North Lexington Avenue, St. Paul, MN, USA.

- [16] DAWSON, J. R. 1976 Digital calculators then and now. An Introduction to Computers in Engineering, Joint Civil Engineering Program Applications -- American Consulting Engineers Council National Convention, Washington, D.C., USA, May, 1976 (In: BROWN [6])
- [17] DISCOTECH MICROCOMPUTER PRODUCTS. 1983 Information on engineering software, Discotech Microcomputer Products, 600 B Street, P. O. Box 1659, Santa Rosa, CA, USA
- [18] FIRMIN, A., M. I. GILMOR, et R. A. COLLINS 1977. Computer programs for structural analysis and design -- a Canadian view. Canadian Journal of Civil Engineering, **4**(77), pp 72-85.
- [19] FREDRIKSSON, B., J. MACKERLE, et B. G. A. PERSSON 1981. Finite element programs in integrated software for structural mechanics and CAD. Computer-Aided Design, **13**(1), pp 27-39.
- [20] FREE, J. 1983. Bits and bytes. Popular Science, **223**(2), pp. 110-112
- [21] GATTAS, M. et J. F. ABEL 1983 Three-dimensional linear dynamic analysis of buildings with 32-bit virtual-memory minicomputers. Computers and Structures, **17**(1), pp 97-104
- [22] HARTLEY, G. A. et J. A. GOLDAK. 1983. Computer educational requirements of a civil or mechanical engineering student. Carleton University Memorandum, March 8, 1983, 2 p
- [23] HOOT, J. E. 1983 Voice lab part 1: a system for speech-recognition computers. Byte, **8**(7), pp. 186-206
- [24] HONEYWELL INFORMATION SYSTEMS INC 1980 Control program-six (CP-6) FORTRAN 77 reference manual
- [25] HUMAR, J. L. et E. W. WRIGHT Computer programs for building analysis and design. Internal Report, Department of Civil Engineering, Ottawa, Ont., Canada, 26 p
- [26] HUQ, M. M. 1978. Computer-aided design of reinforced concrete building structures. Master's Thesis, Carleton University, Ottawa, Ont., Canada.
- [27] INGRAFFEA, A. R., F. K. KULHAWY et J. F. ABEL 1981 Interactive computer graphics for analysis of geotechnical structures. Proceedings of the First International Conference on Computing in Civil Engineering, New York, NY, USA, May 12-14, 1981, pp 864-875
- [28] JANSSEN, J. G. et E. T. MANNING. 1980 In-house preprocessors for ICES STRUDL. Proceedings of the Second Conference on Computing in Civil Engineering, Baltimore, MD, USA June 9-13, 1980, pp 670-677.
- [29] KNUTH, D. E. 1984 The TeXbook. Addison-Wesley Publishing Company, Don Mills Ont. Canada
- [30] LABRECQUE, M. 1983. The tantalizing quest for speech-recognition computers. Popular Science, **223**(1), pp 61-63
- [31] MATEOSIAN, R. 1984 1984, the year of the 32-bit microprocessor. Byte, **9**(1), pp 134-150
- [32] MICROPRO INTERNATIONAL CORP 1983 WORDSTAR user's reference manual. MicroPro International Corp., 33 San Pablo Ave., San Rafael, CA, USA
- [33] MICROSTUF INC 1983. CROSSTALK smart terminal and file transfer program for CP/M Version 2.0. Microstuf Inc., 1845 The Exchange, Suite 205, Atlanta, GA, USA, 53 p

- [34] MUFTI, A. A. Design, availability, and use of engineering computer programs. Lecture Notes, Department of Computer Services, Technical University of Nova Scotia, Halifax, N. S., Canada, 36 p
- [35] PESQUERA, C. I., W. MCGUIRE, et J. F. ABEL. 1983. Interactive graphical preprocessing of three-dimensional framed structures. *Computers and Structures*, 17(1), pp. 1-12
- [36] PROFILE OF APPLICATION PROGRAMS. Multiple Access Division of Canada Systems Group, 307-60 Queen St., Ottawa, Ont., Canada, 4 p
- [37] ROONEY, M. F. et S. E. SMITH. 1983. Artificial intelligence in engineering design. *Computers and Structures*, 16(1-4), pp. 279-288
- [38] SMOLIRA, M. 1975. *Analysis of Tall Buildings by the Force-Displacement Method*. John Wiley and Sons, New York, NY, USA
- [39] SUPERSOFT CORPORATION. 1983. Supersoft FORTRAN and RATFOR for 8086, 8088, 8087, and Z80. Supersoft Corporation, P. O. Box 1628, Champaign, IL, USA. 4 p
- [40] SWANSON, J. A. 1979. Present trends in computerized structural analysis. *Computers and Structures*, 10, pp. 33-37
- [41] THIGPENN, J. A. 1981. Microcomputers in small and medium-sized firms. *Proceedings of the First International Conference on Computing in Civil Engineering*, New York, NY, USA, May 12-14, 1981, pp. 108-120
- [42] WHITE, C. M. 1984. Speech recognition: an idea whose time is coming. *Byte*, 9(1), pp. 213-222
- [43] WILD HEERBRUGG LTD. 1983. Interactive graphics and surveying system GEOMAP. Heerbrugg, Switzerland, 5 p
- [44] WRIGHT, E. W. 1981. *Applications of Computer Graphics in Civil Engineering*. The Canadian Society for Civil Engineering, 2050 Mansfield, Montreal, Que., Canada, 75 p
- [45] WRIGHT, E. W. 1978. EQF: earthquake effects on multi-story building structures program user's manual, preliminary edition. Technology Branch, Public Works, Ottawa, Ont., Canada, 11 p
- [46] WRIGHT, E. W. 1977. The use of conversational computer programs in the structural engineering office. *Canadian Journal of Civil Engineering*, 4, pp. 417-435

Appendix A

Program listing of preprocessor

```

1 - C *** WALL - MAIN *** P. KALAE, VERSION 2.0, NOVEMBER 1, 1983. ***
2 - C MAIN PROGRAM WHICH PREPARES THE DATA FILE FOR STARDYNE ANALYSIS OF A
3 - C CONCRETE COUPLED SHEAR WALL SYSTEM.
4 - C
5 - C * VARIABLES USED - ARRAYS *
6 - C
7 - C RT(50)   = wall thickness at each storey (m)
8 - C WH(50)   = wall height from floor base to top of beam (m) later
9 - C           changed to exclude height of beam
10 - C BH(50)   = beam height (m)
11 - C UDL(50)  = vertical UDL acting on each storey (kN/m)
12 - C PLOAD(50) = concentrated lateral force acting on each storey (kN)
13 - C
14 - C * VARIABLES USED - CHARACTER *
15 - C
16 - C SFILE*50  = name of STARDYNE input data file
17 - C CFILE*50  = name of postprocessor input data file
18 - C RESP*1    = variable checking "Y"/"N" input
19 - C LINE*72   = general buffer for reading input from terminal and
20 - C           passing data to CN72
21 - C
22 - C * VARIABLES USED - DOUBLE PRECISION *
23 - C
24 - C DEN      = concrete density read as kg/cu,m converted to kN/cu,m
25 - C STR      = concrete strength read as MPa converted to kN/cu,m
26 - C E        = modulus of elasticity calculated from DEN and STR
27 - C W1       = width of Wall 1 (m)
28 - C GAP      = distance between walls (m)
29 - C W2       = width of Wall 2 (m)
30 - C BLOAL     = magnitude of lateral UDL acting at base of Wall 1 (kN/m)
31 - C TLOAL     = magnitude of lateral UDL acting at top of Wall 1 (kN/m)
32 - C AR        = maximum allowable aspect ratio (10.0 by default but may be
33 - C           changed)
34 - C RMAX      = actual maximum aspect ratio used after adjusting NEV, NEH1, NEH2
35 - C HINC1     = horizontal length of elements in Wall 1 (m)
36 - C HINC2     = horizontal length of elements in Wall 2 (m)
37 - C VINC      = vertical length of elements in wall excluding beam
38 - C BINC      = vertical length of the 2 beam elements
39 - C X,Y,Z     = nodal coordinates
40 - C
41 - C * VARIABLES USED - INTEGER *
42 - C
43 - C MAXF      = maximum number of floors in building (50) which must
44 - C           correspond to indices in DIMENSION statements should
45 - C           program capacity be changed
46 - C NFI      = actual number of floors in building
47 - C NEH      = nominal number of elements in horizontal direction in Wall 1
48 - C           (4 by default but may be changed) used by ASPECT to calculate
49 - C           NEV, NEH1, and NEH2 based on AR
50 - C NEH1     = actual number of elements in horizontal direction in Wall 1
51 - C NEH2     = actual number of elements in horizontal direction in Wall 2
52 - C NEV      = number of elements in vertical direction for each storey
53 - C           (excluding 2 elements for beam height)
54 - C NODE     = number of nodes in structure
55 - C NW2      = node number where Wall 2 starts
56 - C IEL      = number of elements in structure
57 - C IER      = argument passed back by CN72 which, if positive, indicates
58 - C           the number of correct numeric fields in character string, and
59 - C           if negative, indicates field in which illegal character
60 - C           occurs
61 - C
62 - C * FLAGS USED *
63 - C

```

```

14 - C IJLAL = 1 - if 1 or value before any data is entered. This allows
15 - C          geometry prompts to be bypassed and data is input from
16 - C          editing menu
17 - C          = 1 - set once data is entered or edited in PG
18 - C          = -1 - geometry data cannot be edited
19 - C IJLAL = same as IJLAL but for load data
20 - C          = 1 - normal program execution is followed
21 - C          = 1 - geometry or load prompts are bypassed
22 - C          = -1 - last prompt is displayed again and data is re-entered
23 - C IJLAL = 1 - both geometry and load data are processed
24 - C          = 1 - only geometry data is processed
25 - C
26 - C ** Other variables not defined here or in the STARDYNE manual are
27 - C ** general counters, temporary variables, or their use is evident
28 - C ** from the source code.
29 - C
30 - C * I/O DEVICE NUMBERS *
31 - C
32 - C IUT = 106 - print to terminal
33 - C IUT = 107 - read from terminal
34 - C IUT = 1 - STARDYNE data file
35 - C IUT = 2 - postprocessor data file
36 - C
37 - C * SUBROUTINES CALLED *
38 - C
39 - C CHECK = used to open files and check if files already exist or if
40 - C          any other errors are encountered
41 - C          C = prints prompt, reads answer, and checks for any errors
42 - C          in input (illegal alphanumeric input or too few arguments),
43 - C          or executes special commands GEOM, LOAD, and STOP
44 - C          WL = called by Q to determine number of characters in prompt so
45 - C          it is printed correctly
46 - C          CN72 = reads alphanumeric string (max. of 72 characters) and returns
47 - C          the number of valid numeric fields (excluding D format) or
48 - C          the field in which illegal character was encountered
49 - C          PG = displays geometry data and allows user to edit it
50 - C          PL = displays load data and allows user to edit it
51 - C          ASPECT = calculates the aspect ratio of elements based on AR and
52 - C          NEH, and returns NEV, NEH1, and NEH2
53 - C          GEOM = calculates and writes geometry input data for STARDYNE
54 - C          LOAD = calculates and writes load input data for STARDYNE
55 - C
56 - C * COMMON STATEMENTS AND SUBROUTINES SHARING THEM *
57 - C
58 - C GEOM1 - MAIN,PG,LOAD
59 - C GEOM2 - MAIN,PG,GEOM,LOAD
60 - C GEOM3 - MAIN,ASPECT
61 - C REST1 - MAIN,PG,GEOM
62 - C FILES - MAIN,GEOM,LOAD
63 - C ALL1 - MAIN,PG,PL
64 - C LOAD1 - MAIN,PL,LOAD
65 - C Q1 - MAIN,PG,PL
66 - C AR1 - ASPECT,GEOM,LOAD
67 - C
68 - C * RESTRICTIONS *
69 - C
70 - C 1. Data must be entered in the units specified.
71 - C 2. Only coupled shear walls may be analyzed such that Wall 1, on
72 - C    which the lateral loads act, is wider than or as wide as Wall 2.
73 - C 3. At least some openings must exist between the walls (GAP is greater than 0,
74 - C    WH(1) is greater than BH(1))
75 - C 4. Present dimensioning of program allows structures of at most 50
76 - C    stories to be analyzed. This may be increased by changing the
77 - C    DIMENSION statements, MAXF value, and appropriate formats.
78 - C
79 - C *****
80 - C
81 - C
82 - C
83 - C
84 - C
85 - C
86 - C
87 - C
88 - C
89 - C
90 - C
91 - C
92 - C
93 - C
94 - C
95 - C
96 - C
97 - C
98 - C
99 - C
100 - C
101 - C
102 - C
103 - C
104 - C
105 - C
106 - C
107 - C
108 - C
109 - C
110 - C
111 - C
112 - C
113 - C
114 - C
115 - C
116 - C
117 - C
118 - C
119 - C
120 - C
121 - C
122 - C
123 - C
124 - C
125 - C
126 - C
127 - C
128 - C
129 - C
130 - C
131 - C
132 - C

```

*** === D I S C L A I M E R === ***

```

133 - C ALTHOUGH THIS PROGRAM INCLUDES MANY TESTS TO CATCH INCORRECT DATA, AND
134 - C ALTHOUGH THE PROGRAM HAS BEEN RIGOROUSLY TESTED, IT HAS BEEN INTENDED
135 - C ONLY FOR ILLUSTRATION PURPOSES, AND NEITHER THE AUTHOR, THE DEPARTMENT
136 - C OF CIVIL ENGINEERING, NOR CARLETON UNIVERSITY ASSUME ANY RESPONSIBILI-
137 - C TY FOR ITS USE BY ANY INDIVIDUAL OR ORGANIZATION.
138 - C SINCE LARGE QUANTITIES OF DATA MUST BE TRANSMITTED OVER PUBLIC COMMU-
139 - C NICATION LINES, NOISE MAY BE INTRODUCED AND MAKE THE DATA USELESS EVEN
140 - C THOUGH IT WAS ENTERED AND PROCESSED CORRECTLY. IT IS THE ENGINEER'S
141 - C RESPONSIBILITY TO ENSURE THAT ALL DATA IS CORRECT.
142 - C
143 - C *****
144 - C
145 - C
146 - C      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
147 - C      IMPLICIT INTEGER(I-N)
148 - C      CHARACTER LINE*72,RESP*1,SFILE*36,CFILE*36
149 - C      DIMENSION PT(50),PLOAD(50),WH(50),BH(50),UDL(50)
150 - C      DATA NFL,MAXF/1,50/
151 - C      DATA DEN,STR,E,W1,GAP,W2/2400.00,25.00,25200.00,3*0.00/
152 - C      DATA BLOAD,TLOAD,AR/2*0.00,10.00/
153 - C      DATA NEH,NFL/4,1/
154 - C      COMMON /GEOM1/ DEN,STR,E,W1,GAP,W2,AR,NEH
155 - C      COMMON /GEOM2/ PT,WH,BH
156 - C      COMMON /GEOM3/ WMAX,WMIN,BMAX,BMIN
157 - C      COMMON /REST1/ NODE,NW2,IGEOM,IEG
158 - C      COMMON /FILES/ SFILE,CFILE
159 - C      COMMON /ALL1/ NFL,MAXF
160 - C      COMMON /LOAD1/ UDL,BLOAD,TLOAD,PLOAD
161 - C      COMMON /Q1/ IGF1AG,ILF1AG,IGO,IGF2
162 - C
163 - C      INITIALIZE VARIABLES AND OPEN FILES
164 - C
165 - C      WRITE(108,50)
166 - C      DO 20 I=1,MAXF
167 - C          PLOAD(I)=0.00
168 - C          WH(I)=0.00
169 - C          BH(I)=0.00
170 - C          PT(I)=0.00
171 - C          UDL(I)=1.00-99
172 - C      20 CONTINUE
173 - C      30 CONTINUE
174 - C          IGF2=0
175 - C          IGF1AG=0
176 - C          ILF1AG=0
177 - C          IGO=0
178 - C          IPASS=0
179 - C      50 FORMAT(/
180 - C          + ' STARDYNE PREPROCESSOR FOR COUPLED SHEAR WALLS - METRIC VERSION'
181 - C          + /1X,62'='//
182 - C          + ' Enter numerical data after each prompt in the appropriate units'
183 - C          + ' separated by spaces or commas.'//
184 - C          + ' Any of the following commands override normal execution:/'
185 - C          + ' STOP = Stops program immediately/'
186 - C          + ' GEOM = Geometry parameters are displayed and may be altered/'
187 - C          + ' LOAD = Load parameters are displayed and may be altered'//)
188 - C      55 CONTINUE
189 - C          PRINT *, 'Is this a "GEOMETRY ONLY" run?'
190 - C          READ '(A1)', RESP
191 - C          IF (RESP.EQ.'Y') THEN
192 - C              IGEOM=1
193 - C              ILF1AG=-1
194 - C          ELSE IF (RESP.EQ.'N') THEN
195 - C              IGEOM=0
196 - C          ELSE IF (RESP.EQ.'S') THEN
197 - C              PRINT *, '== RUN ABORTED =='
198 - C              STOP
199 - C          ELSE
200 - C              PRINT *, 'Please answer "Y", "N", or "S" to abort.'
201 - C              GOTO 55

```

```

202 -     ENCL IF
203 -     PRINT *, 'Enter name of file where Stardyne input will be stored '
204 -     READ(105,80) SFILE
205 -     IF(SFILE(1:4).EQ.'STOP')
206 -     + PRINT *, '== RUN ABORTED ==', STOP
207 - 6.  FORMAT(A36)
208 -     CALL OCHECK(1,SFILE,'NEW','SEQUENTIAL',IER)
209 -     IF(IER.EQ.1) STOP
210 -     IF(IGEQ.EQ.1) GOTO 100
211 -     PRINT *,
212 -     + 'Enter name of file where postprocessor commands will be stored '
213 - 9.  CONTINUE
214 -     READ(105,80) CFILE
215 -     IF(CFILE(1:4).EQ.'STOP')
216 -     + PRINT *, '== RUN ABORTED ==', STOP
217 -     IF(CFILE.EQ.SFILE) THEN
218 -     PRINT *, 'You cannot use this file name - choose another '
219 -     GOTO 90
220 -     ELSE
221 -     CALL OCHECK(2,CFILE,'NEW','SEQUENTIAL',IER)
222 -     IF(IER.EQ.1) STOP
223 -     ENCL IF
224 - 10. CONTINUE
225 -     PRINT *, 'Enter the following concrete parameters in proper units '
226 -     CALL Q('Density (kg/cu.m), Strength (MPa):',2,LINE)
227 -     IF(IGO.EQ.-1) IGO=0; GOTO 100
228 -     IF(IGO.EQ.1) GOTO 300
229 -     READ(LINE,120) DEN,STR
230 -     E=DEN**1.500*0.04300*DSQRT(STR)
231 - 120 FORMAT(6G)
232 -     IF(IPASS.GT.0) GOTO 300
233 - 130 CONTINUE
234 -     PRINT *, 'Enter the following widths in metres:'
235 -     CALL Q('Wall 1, distance between walls, Wall 2:',3,LINE)
236 -     IF(IGO.EQ.-1) IGO=0; GOTO 130
237 -     IF(IGO.EQ.1) GOTO 300
238 -     READ(LINE,120) W1,GAP,W2
239 -     IF(IPASS.GT.0) GOTO 300
240 - 150 CONTINUE
241 -     CALL Q('Enter the number of floors in building:',1,LINE)
242 -     IF(IGO.EQ.-1) IGO=0; GOTO 150
243 -     IF(IGO.EQ.1) GOTO 300
244 -     READ(LINE,120) NFL
245 -     IF((NFL.LE.0).OR.(NFL.GT.MAXF)).PRINT *,
246 -     + 'ERROR - maximum number of floors exceeded:',NFL, GOTO 150
247 -     IF(IPASS.GT.0) GOTO 300
248 - 160 CONTINUE
249 -     PRINT *,
250 -     + 'For the first floor, enter the following data in metres:'
251 -     CALL Q('Wall thickness, storey height, beam height:',3,LINE)
252 -     IF(IGO.EQ.-1) IGO=0; GOTO 160
253 -     IF(IGO.EQ.1) GOTO 300
254 -     READ(LINE,120) PT(1),WH(1),BH(1)
255 -     IF((PT(1).LE.0.DO).OR.(WH(1).LE.0.DO).OR.(BH(1).LE.0.DO)) PRINT *,
256 -     + 'ERROR - All values must be greater than zero.'; GOTO 160
257 -     IF(WH(1).LT.BH(1)) PRINT *,
258 -     + 'ERROR - Beam height must be less than storey height.'; GOTO 160
259 -     IF(IPASS.GT.0) GOTO 300
260 - 170 CONTINUE
261 -     IF(NFL.LT.2) GOTO 300
262 -     WRITE(108,180) NFL
263 - 180 FORMAT(' Do storeys 2 to ',I3,
264 -     + ' have constant floor height, beam height and wall thickness?')
265 -     READ '(A)',RESP
266 -     IF(RESP.EQ.'Y') THEN
267 - 185 CONTINUE
268 -     PRINT *, 'Is the geometry the same as for first storey (Y/N)?'
269 -     READ '(A)',RESP
270 -     IF(RESP.EQ.'Y') THEN

```

```

271 -      TH=PT(1)
272 -      TWH=WH(1)
273 -      TBH=BH(1)
274 -      ELSE IF(RESPEQ,'S') THEN
275 -          PRINT *, '== RUN ABORTED =='
276 -          STOP
277 -      ELSE IF(RESPEQ,'N') THEN
278 - 190  CONTINUE
279 -      CALL Q('Enter wall thickness, storey height, beam height:',
280 -          +      3,LINE)
281 -      IF(IGO,EQ,-1) IGO=0; GOTO 190
282 -      IF(IGO,EQ,1) GOTO 300
283 -      READ(LINE,120) TH,TWH,TBH
284 -      IF((TH,LE,0,DO).OR.(TWH,LE,0,DO).OR.(TBH,LE,0,DO)) PRINT *,
285 -          +      'ERROR - All values must be positive.'; GOTO 190
286 -      IF(TWH,LT,TBH) PRINT *,
287 -          +      'ERROR - Beam height must be less than storey height.'; GOTO 190
288 -      ELSE
289 -          PRINT *, 'Please answer "Y", "N", or "S" to abort.'
290 -          GOTO 185
291 -      END IF
292 -      DO 200 I=2,NFL
293 -          PT(I)=TH
294 -          WH(I)=TWH
295 -          BH(I)=TBH
296 - 200  CONTINUE
297 -      ELSE IF(RESPEQ,'N') THEN
298 -          WRITE(108,210) NFL
299 - 210  FORMAT(
300 -          +      'For each storey where the wall thickness, floor height, or/'
301 -          +      'beam height changes, enter the storey (2-',12,
302 -          +      ') and the three/'
303 -          +      'parameters in metres ( greater than 0). End with a blank line.')
304 - 220  CONTINUE
305 -      CALL Q(' ',4,LINE)
306 -      IF(IGO,EQ,-1) IGO=0; GOTO 220
307 -      IF(IGO,EQ,1) GOTO 300
308 -      CALL CN72(LINE,IER)
309 -      IF(IER,EQ,0) GOTO 250
310 -      READ(LINE,120) I,TH,TWH,TBH
311 -      IF((I,LT,2).OR.(I,GT,NFL)) PRINT *,
312 -          +      'ERROR - Storey out of range:',I; GOTO 220
313 -      IF((TH,LE,0,DO).OR.(TWH,LE,0,DO).OR.(TBH,LE,0,DO)) PRINT *,
314 -          +      'ERROR - All values must be greater than zero. Try again:'
315 -          +      GOTO 220
316 -      IF(TBH,GT,TWH) PRINT *,
317 -          +      'ERROR - Beam height must be less than storey height. Try again:'
318 -          +      ; GOTO 220
319 -      PT(I)=TH
320 -      WH(I)=TWH
321 -      BH(I)=TBH
322 -      GOTO 220
323 - 250  CONTINUE
324 -      DO 260 I=2,NFL
325 -          IF((PT(I),LE,0,DO).OR.(WH(I),LE,0,DO).OR.(BH(I),LE,0,DO)) THEN
326 -              PT(I)=PT(I-1)
327 -              WH(I)=WH(I-1)
328 -              BH(I)=BH(I-1)
329 -          ELSE
330 -              END IF
331 - 260  CONTINUE
332 -      ELSE IF(RESPEQ,'S') THEN
333 -          PRINT *, '== RUN ABORTED =='
334 -          STOP
335 -      ELSE
336 -          PRINT *, 'Please answer "Y", "N", or "S" to abort.'
337 -          GOTO 170
338 -      END IF
339 - 300  CONTINUE

```



```

340 -      IPASS=1
341 -      IFLAG=1
342 -      IF(IGO.EQ.0) CALL PG(108)
343 -      IGO=0
344 - C
345 - C TEST TO MAKE SURE PARAMETERS ARE IN PROPER RANGE
346 - C
347 -      IF((DEN.LE.0.D0).OR.(STR.LE.0.D0).OR.(E.LE.0.D0)) PRINT *,
348 - + 'ERROR - Concrete density, strength, and E must be greater than 0', GOTO 10w
349 -      IF((W1.LE.0.D0).OR.(GAP.LE.0.D0).OR.(W2.LE.0.D0)) PRINT *,
350 - + 'ERROR - All widths must be greater than zero.', GOTO 130
351 -      IF(W1.LT.W2) THEN
352 -          PRINT *, 'ERROR - Wall 1 must be wider than Wall 2. Swapping.'
353 -          TLOC=W2
354 -          W2=W1
355 -          W1=TLOC
356 -          GOTO 300
357 -      ELSE
358 -          END IF
359 -      IF((NFL.LE.0).OR.(NFL.GT.MAXF)) PRINT *,
360 - + 'ERROR - Building must have 1 to', MAXF, ' storeys.', GOTO 150
361 -      IER=0
362 -      DO 310 I=1,NFL
363 -          IF((PT(I).LE.0.D0).OR.(WH(I).LE.0.D0).OR.(BH(I).LE.0.D0)) THEN
364 -              WRITE(108,305) I
365 -          305      FORMAT(' ERROR in data for floor',I3,' :')
366 -              PRINT *,
367 - + 'Wall thickness, storey height, and beam height must be greater than 0.'
368 -              IER=IER+1
369 -          ELSE
370 -              END IF
371 -          IF(WH(I).LT.BH(I)) THEN
372 -              WRITE(108,305) I
373 -              PRINT *,
374 - + 'Storey height must be greater than beam height.'
375 -              IER=IER+1
376 -          ELSE
377 -              END IF
378 -          310      CONTINUE
379 -          IF(IER.NE.0) THEN
380 -              PRINT *, 'There were', IER, ' errors which must be corrected.'
381 -              GOTO 160
382 -          ELSE
383 -              END IF
384 -          WMAX=WH(I)-BH(I)
385 -          WMIN=WMAX
386 -          BMAX=BH(I)
387 -          BMIN=BMAX
388 -          DO 312 I=1,NFL
389 -              CH=WH(I)-BH(I)
390 -              IF(CH.GT.WMAX) WMAX=CH
391 -              IF(CH.LT.WMIN) WMIN=CH
392 -              IF(BH(I).GT.BMAX) BMAX=BH(I)
393 -              IF(BH(I).LT.BMIN) BMIN=BH(I)
394 -          312      CONTINUE
395 -          IWARN=0
396 -          RATIO=WMAX/WMIN
397 -          IF(RATIO.GT.5.D0) WRITE(108,313) RATIO; IWARN=1
398 -          RATIO=BMAX/BMIN
399 -          IF(RATIO.GT.10.D0) WRITE(108,314) RATIO; IWARN=1
400 -          RATIO=WMAX/BMIN
401 -          IF(RATIO.GT.35.D0) WRITE(108,315) RATIO; IWARN=1
402 -          313      FORMAT(
403 - + ' WARNING - Wall height ratio MAX/MIN appears too large:',F6.1)
404 -          314      FORMAT(
405 - + ' WARNING - Beam height ratio MAX/MIN appears too large:',F6.1)
406 -          315      FORMAT(
407 - + ' WARNING - Height ratio WALL MAX/BEAM MIN appears too large:',
408 - + F6.1)

```

```

409 - IF (IWARN.GT.0) THEN
410 - 316 CONTINUE
411 - PRINT *,
412 - + 'Mesh may be impossible to generate or may give wrong results.'
413 - PRINT *, 'Do you wish to proceed (Y/N)?'
414 - READ '(A1)', RESP
415 - IF (RESP.EQ.'Y') THEN
416 - ELSE IF (RESP.EQ.'N') THEN
417 - GOTO 300
418 - ELSE IF (RESP.EQ.'S') THEN
419 - PRINT *, '== RUN ABORTED =='
420 - STOP
421 - ELSE
422 - PRINT *, 'Please answer "Y", "N", or "S" to abort.'
423 - GOTO 316
424 - END IF
425 - ELSE
426 - END IF
427 - C
428 - C GEOMETRY DATA HAS BEEN ENTERED CORRECTLY. NOW ENTER LOAD DATA.
429 - C
430 - IFLAG=-1
431 - IF (ILFLAG.NE.0) GOTO 500
432 - 320 CONTINUE
433 - WRITE(108,322) NFL
434 - 322 FORMAT(
435 - + ' VERTICAL UDL - Positive value indicates downward force.'/
436 - + ' Enter the storey no. (1-12,'
437 - + ' and the load magnitude (kN/m)'/
438 - + ' where the load changes. End with a blank line.')
439 - 324 CONTINUE
440 - CALL Q(' ',2,LINE)
441 - IF (IGO.EQ.-1) IGO=0; GOTO 320
442 - IF (IGO.EQ.1) GOTO 500
443 - CALL CN72(LINE,IER)
444 - IF (IER.EQ.0) GOTO 326
445 - READ(LINE,120) I,TLOAD
446 - IF ((I.LT.1).OR.(I.GT.NFL)) PRINT *,
447 - + 'ERROR - Storey out of range:',I,' Try again:'; GOTO 324
448 - UDL(I)=TLOAD
449 - GOTO 324
450 - 326 CONTINUE
451 - DO 328 I=2,NFL
452 - IF (UDL(I).EQ.1.00-99) UDL(I)=UDL(I-1)
453 - 328 CONTINUE
454 - 330 CONTINUE
455 - PRINT *,
456 - + 'LATERAL LOAD - Positive value indicates force pushing on wall.'
457 - PRINT *, 'For linearly varying lateral load on wall 1 (in kN/m)'
458 - CALL Q('Enter magnitude at bottom, magnitude at top:',2,LINE)
459 - IF (IGO.EQ.-1) IGO=0; GOTO 330
460 - IF (IGO.EQ.1) GOTO 500
461 - READ(LINE,120) BLOAD,TLOAD
462 - 340 CONTINUE
463 - WRITE(108,350) NFL
464 - 350 FORMAT(' Shear walls are connected at',I4,' storeys.'/
465 - + ' Enter the storey no. and the point load in kN acting at top'/
466 - + ' of storey. End with a blank line.')
467 - 360 CONTINUE
468 - CALL Q(' ',2,LINE)
469 - IF (IGO.EQ.-1) IGO=0; GOTO 340
470 - IF (IGO.EQ.1) GOTO 500
471 - READ(LINE,120) J,TLOC
472 - IF ((J.GT.0).AND.(J.LE.NFL)) THEN
473 - PLOAD(J)=TLOC
474 - GOTO 360
475 - ELSE
476 - END IF
477 - ILFLAG=1

```

```

478 - 500 CONTINUE
479 - IF((IGO.EQ.0).AND.(IFLAG.NE.-1)) CALL PL(106)
480 - IGO=0
481 - IF((IFLAG.NE.-1) THEN
482 - 505 CONTINUE
483 - PRINT *, 'Do you wish to change geometry data?'
484 - READ '(A1)', RESP
485 - IF(RESQ.EQ.'Y') THEN
486 - IFLAG=1
487 - GOTO 300
488 - ELSE IF(RESQ.EQ.'S') THEN
489 - PRINT *, '== RUN ABORTED =='
490 - STOP
491 - ELSE IF(RESQ.EQ.'N') THEN
492 - CONTINUE
493 - ELSE
494 - PRINT *, 'Please answer "Y", "N", or "S" to abort.'
495 - GOTO 505
496 - END IF
497 - ELSE
498 - END IF
499 - IF(IGEOM.EQ.0) THEN
500 - REWIND 2
501 - CALL PG(2)
502 - CALL PL(2)
503 - ELSE
504 - END IF
505 - C
506 - C ALL DATA HAVE BEEN ENTERED AND WRITTEN ON FILES.
507 - C
508 - CALL ASPECT(W1,W2,AP,NEH,IGEOM,IER)
509 - IF(IER.NE.0) GOTO 300
510 - REWIND 1
511 - CALL GEOM(DEN,E,W1,GAP,W2,NFL,IEL)
512 - IF(IEG.NE.0) GOTO 300
513 - IF(IGEOM.EQ.0) CALL LOAD(NFL,NODE,NW2,IEL)
514 - 510 CONTINUE
515 - WRITE(108,520)
516 - 520 FORMAT(///' Do you wish to process a new case (Y/N)?')
517 - READ '(A1)', RESP
518 - IF(RESQ.EQ.'Y') THEN
519 - E=E/1.D3
520 - DEN=DEN/9.810-3
521 - DO 530 I=1,NFL
522 - WH(I)=WH(I)+BH(I)
523 - 530 CONTINUE
524 - GOTO 30
525 - ELSE IF((RESQ.EQ.'N').OR.(RESQ.EQ.'S')) THEN
526 - STOP
527 - ELSE
528 - PRINT *, 'Please answer "Y" or "N":'
529 - GOTO 510
530 - END IF
531 - END
532 - C
533 - C *****
534 - C **** PG **** P. KALAB, VERSION 2.0, NOVEMBER 16, 1983, ****
535 - C SUBROUTINE DISPLAYS GEOMETRY CONTROL PARAMETERS AND ALLOWS USER TO
536 - C MAKE ANY CHANGES.
537 - C * SYNTAX * PG(NDEV)
538 - C * NDEV = DEVICE NO. WHERE DATA WILL BE PRINTED
539 - C * CALL BY * WALL, Q
540 - C * CALLS * Q
541 - C * COMMON * GEOM1(WALL), GEOM2(WALL,LOAD), REST1(WALL),
542 - C * ALL1(WALL,ASPECT,LOAD,PL), Q1(WALL,Q)
543 - C
544 - SUBROUTINE PG(NDEV)
545 - IMPLICIT DOUBLE PRECISION(A-H,O-Z)
546 - IMPLICIT INTEGER(I-N)

```

```

1237 -      GOTO 170
1238 -      ELSE
1239 -      NEV=NEV+1
1240 -      GOTO 140
1241 -      END IF
1242 -      ELSE
1243 -      IF(R.GT,RMAX) RMAX=R
1244 -      ICOUNT=0
1245 -      END IF
1246 - C
1247 - C HIGH AND NARROW WALL 2
1248 - C
1249 - 170 CONTINUE
1250 -      HINC2=W2/DFLOAT(NEH2)
1251 -      VINC=WMAX/DFLOAT(NEV)
1252 -      R=VINC/HINC2
1253 -      IF(R.GT,AR) THEN
1254 -      ICOUNT=ICOUNT+1
1255 -      IF(ICOUNT.GT,100) THEN
1256 -      WRITE(108,180) R,NEV
1257 - 180      FORMAT(' Ratio WMAX/HINC2 =',F6.1,' NEV =', 14)
1258 -      ICOUNT=0
1259 -      IER=6
1260 -      GOTO 200
1261 -      ELSE
1262 -      NEV=NEV+1
1263 -      GOTO 170
1264 -      END IF
1265 -      ELSE
1266 -      IF(R.GT,RMAX) RMAX=R
1267 -      ICOUNT=0
1268 -      END IF
1269 - C
1270 - C HIGH BEAM IN WALL 1
1271 - C
1272 - 200 CONTINUE
1273 -      HINC1=W1/DFLOAT(NEH1)
1274 -      R=(BMAX/2,DO)/HINC1
1275 -      IF(R.GT,AR) THEN
1276 -      IER=7
1277 -      WRITE(108,210) R,BMAX,NEH1
1278 - 210      FORMAT(' Desired geometry exceeds limits of this program.'/
1279 -      +' Aspect ratio =',F6.1,' Beam height =',F9.1,
1280 -      +' No. of hor. elem. in Wall 1 =',16/
1281 -      +' Either beam is too deep or mesh is too fine.')
1282 -      ELSE
1283 -      IF(R.GT,RMAX) RMAX=R
1284 -      END IF
1285 - C
1286 - C HIGH BEAM IN WALL 2
1287 - C
1288 -      HINC2=W2/DFLOAT(NEH2)
1289 -      R=(BMAX/2,DO)/HINC2
1290 -      IF(R.GT,AR) THEN
1291 -      IER=8
1292 -      WRITE(108,230) R,BMAX,NEH2
1293 - 230      FORMAT(' Desired geometry exceeds limits of this program.'/
1294 -      +' Aspect ratio =',F6.1,' Beam height =',F9.1,
1295 -      +' No. of hor. elem. in Wall 2 =',16/
1296 -      +' Either beam is too deep or mesh is too fine.')
1297 -      ELSE
1298 -      IF(R.GT,RMAX) RMAX=R
1299 -      END IF
1300 - C
1301 - C MAKE SECOND PASS TO MAKE SURE ALL ASPECT RATIOS ARE STILL OK
1302 - C
1303 -      IF(IPASS.EQ,0) THEN
1304 -      IER=0
1305 -      IPASS=1

```

```

616 - PRINT *, '== RUN ABORTED =='; STOP
617 - ELSE
618 - PRINT *, 'Please answer "Y", "N", or "S" to abort.'
619 - GOTO 115
620 - END IF
621 - ELSE
622 - IGO=0
623 - 125 CONTINUE
624 - WRITE(108,80)
625 - READ(105,50) RESP
626 - IF (RESP.EQ.'Y') THEN
627 - ELSE IF (RESP.EQ.'N') THEN,
628 - IGF2=0
629 - RETURN
630 - ELSE IF (RESP.EQ.'S') THEN
631 - PRINT *, '== RUN ABORTED =='
632 - STOP
633 - ELSE
634 - PRINT *, 'Please answer "Y", "N", or "S" to abort.'
635 - GOTO 125
636 - END IF
637 - END IF
638 - 150 CONTINUE
639 - IF (IPASS.NE.0) THEN
640 - IF (ICOUNT.EQ.0) GOTO 300
641 - WRITE(108,10) DEN,STR,E,W1,GAP,W2,NFL,NEH,AR
642 - 160 CONTINUE
643 - WRITE(108,80)
644 - READ(105,50) RESP
645 - IF (RESP.EQ.'Y') THEN
646 - ELSE IF (RESP.EQ.'N') THEN
647 - GOTO 300
648 - ELSE IF (RESP.EQ.'S') THEN
649 - PRINT *, '== RUN ABORTED =='
650 - STOP
651 - ELSE
652 - PRINT *, 'Please answer "Y", "N", or "S" to abort.'
653 - GOTO 160
654 - END IF
655 - ELSE
656 - END IF
657 - WRITE(108,30)
658 - C
659 - C DECISION TABLE FOR FIRST SET OF DATA DISPLAYED (FORMAT 10)
660 - C
661 - 200 CONTINUE
662 - CALL Q(' ',2,LINE)
663 - CALL CN72(LINE,IER)
664 - IF (IER.EQ.0) IPASS=1; GOTO 150
665 - READ(LINE,60) K,DNUM
666 - IF (K.EQ.1) THEN
667 - DEN=DNUM
668 - E=DEN**1.500*0.04300*DSQRT(STR)
669 - ELSE IF (K.EQ.2) THEN
670 - STR=DNUM
671 - E=DEN**1.500*0.04300*DSQRT(STR)
672 - ELSE IF (K.EQ.3) THEN
673 - E=DNUM
674 - ELSE IF (K.EQ.4) THEN
675 - W1=DNUM
676 - ELSE IF (K.EQ.5) THEN
677 - GAP=DNUM
678 - ELSE IF (K.EQ.6) THEN
679 - W2=DNUM
680 - ELSE IF (K.EQ.7) THEN
681 - NFL=IDINT(DNUM)
682 - IF (NFL.LE.0) PRINT *,
683 - + 'WARNING - Building must have at least one storey.'; NFL=1
684 - IF (NFL.GT.MAXF) PRINT *,

```

```

685 -      + 'WARNING - Maximum number of storeys exceeded.', NFL=MAXF
686 -      ELSE IF(K.EQ.8) THEN
687 -          NEH=ABS(IDINT(DNUM))
688 -          IF(NEH.EQ.0) NEH=1
689 -          ELSE IF(K.EQ.9) THEN
690 -              AR=DNUM
691 -              IF(AR.LT.1.D0) AR=1.D0
692 -          ELSE
693 -              PRINT *, 'ERROR - Code no. out of range:', K
694 -          END IF
695 -          ICOUNT=ICOUNT+1
696 -          GOTG 200
697 - C
698 - C DISPLAY SECOND SET OF DATA AND GIVE USER OPTION OF MAKING ANY
699 - C CHANGES
700 - C
701 - 300 CONTINUE
702 -      WRITE(108,15)
703 -      DO 350 I=1,NFL
704 -          WRITE(108,20) I,PT(I),WH(I),BH(I)
705 - 350 CONTINUE
706 -      IF(IPASS.NE.0) THEN
707 - 360 CONTINUE
708 -          WRITE(108,80)
709 -          READ(105,50) RESP
710 -          IF(RESP.EQ.'Y') THEN
711 -              ELSE IF(RESP.EQ.'N') THEN
712 -                  IGF2=0
713 -                  RETURN
714 -              ELSE IF(RESP.EQ.'S') THEN
715 -                  PRINT *, '== RUN ABORTED =='
716 -                  STOP
717 -              ELSE
718 -                  PRINT *, 'Please answer "Y", "N", or "S" to abort.'
719 -                  GOTG 360
720 -              END IF
721 -          ELSE
722 -              END IF
723 -          WRITE(108,40)
724 - 400 CONTINUE
725 -          CALL Q(' ',4,LINE)
726 -          CALL CN72(LINE,IER)
727 -          IF(IER.EQ.0) IPASS=1; GOTG 300
728 -          READ(LINE,60) I,TH,TWH,TBH
729 -          IF((I.LE.0).OR.(I.GT.NFL)) THEN
730 -              PRINT *, 'ERROR - Storey out of range:', I
731 -              PRINT *, 'Try again:'
732 -              ELSE
733 -                  PT(I)=TH
734 -                  WH(I)=TWH
735 -                  BH(I)=TBH
736 -              END IF
737 -              GOTG 400
738 -          END
739 - C
740 - C *****
741 - C **** PL **** P. KALAB; VERSION 1.1; NOVEMBER 1, 1983. ****
742 - C SUBROUTINE WHICH DISPLAYS THE LOAD CONTROL PARAMETERS AND ALLOWS THE
743 - C USER TO MAKE CHANGES.
744 - C * SYNTAX * PL(NDEV)
745 - C          NDEV=DEVICE NUMBER WHERE DATA WILL BE PRINTED
746 - C * CALL BY * WALL, Q
747 - C * CALLS * Q
748 - C * COMMON * LOAD1 (WALL, LOAD)
749 - C          Q1 (WALL, PG)
750 - C
751 - C SUBROUTINE PL(NDEV)
752 - C IMPLICIT DOUBLE PRECISION(A-H,O-Z)
753 - C IMPLICIT INTEGER(I-N)

```

```

754 - CHARACTER RESP*1, LINE#72
755 - DIMENSION PLOAD(50), UDL(50)
756 - COMMON /LOAD1/ UDL, BLOAD, TLOAD, PLOAD
757 - COMMON /ALL1/ NFL, MAXF
758 - COMMON /Q1/ IGFLAG, ILFLAG, IGO, IGF2
759 - 1C FORMAT(/' APPLIED LOADS: '/1X, 14='/'
760 - + ' STOREY VERTICAL UDL LATERAL CONC. LOAD' /
761 - + 12X, '(kN/m)' , 13X, '(kN)' )
762 - 2C FORMAT(15, F13.3, 7X, F11.3)
763 - 3C FORMAT('/ LINEARLY VARYING LATERAL LOAD (kN/m): '/
764 - + 13, ' ) Bottom =', F9.3, ' ; ', 14, ' ) Top =', F9.3)
765 - 4C FORMAT(/
766 - + ' To change any loads, enter the storey no. or code no. ' /
767 - + ' and the new value(s). End with a blank line. ' )
768 - 6C FORMAT(3G)
769 - IF(ILFLAG.EQ.-1) THEN
770 - WRITE(108, 70)
771 - 7C FORMAT(' This is a "GEOMETRY ONLY" run and loads will not',
772 - + ' be processed. ' )
773 - IGO=-1
774 - RETURN
775 - ELSE
776 - IGO=0
777 - END IF
778 - 10C CONTINUE
779 - WRITE(NDEV, 10)
780 - DO 120 I=1, NFL
781 - WRITE(NDEV, 20) I, UDL(I), PLOAD(I)
782 - 12C CONTINUE
783 - WRITE(NDEV, 30) NFL+1, BLOAD, NFL+2, TLOAD
784 - IF(ILFLAG.EQ.0) THEN
785 - 125 CONTINUE
786 - PRINT *, ' '
787 - PRINT *, 'Do you wish to enter loads from this level?'
788 - READ '(A1)', RESP
789 - IF(RESP.EQ.'Y') THEN
790 - ILFLAG=1
791 - IGO=1
792 - WRITE(108, 40)
793 - GOTO 200
794 - ELSE IF(RESP.EQ.'N') THEN
795 - IGO=-1
796 - PRINT *, 'Proceed with normal entry of data.'
797 - PRINT *, ' '
798 - RETURN
799 - ELSE IF(RESP.EQ.'S') THEN
800 - PRINT *, '== RUN ABORTED =='
801 - STOP
802 - ELSE
803 - PRINT *, 'Please answer "Y", "N", or "S" to abort.'
804 - GOTO 125
805 - END IF
806 - ELSE
807 - END IF
808 - IF(NDEV.EQ.108) THEN
809 - 130 CONTINUE
810 - WRITE(108, 140)
811 - 140 FORMAT('/ Do you wish to make any changes?')
812 - READ '(A1)', RESP
813 - IF(RESP.EQ.'Y') THEN
814 - WRITE(108, 40)
815 - ILFLAG=1
816 - GOTO 200
817 - ELSE IF(RESP.EQ.'N') THEN
818 - RETURN
819 - ELSE IF(RESP.EQ.'S') THEN
820 - PRINT *, '== RUN ABORTED =='
821 - STOP
822 - ELSE

```

```

823 -      PRINT *, 'Please answer "Y", "N", or "S" to abort.'
824 -      GOTO 130
825 -      END IF
826 -      ELSE
827 -      RETURN
828 -      END IF
829 - 200 CONTINUE
830 -      CALL Q(' ', 2, LINE)
831 -      READ(LINE, 60) K, DNUM1, DNUM2
832 -      IF((K.EQ.0).AND.(DNUM1.EQ.0.D0)) THEN
833 -      GOTO 100
834 -      ELSE IF(K.EQ.NFL+1) THEN
835 -      BLOAD=DNUM1
836 -      ELSE IF(K.EQ.NFL+2) THEN
837 -      TLOAD=DNUM1
838 -      ELSE IF((K.GE.1).AND.(K.LE.NFL)) THEN
839 -      UDL(K)=DNUM1
840 -      PLOAD(K)=DNUM2
841 -      ELSE
842 -      PRINT *, 'ERROR - Code out of range.', K
843 -      END IF
844 -      GOTO 200
845 -      END
846 - C
847 - C *****
848 - C **** Q **** P. KALAB; VERSION 2.0, OCTOBER 28, 1983. ****
849 - C SUBROUTINE WHICH PRINTS A QUESTION PROMPT FOR THE USER AND READS
850 - C THE ANSWER AS A CHARACTER STRING. IT CHECKS WHETHER THE ANSWER HAS
851 - C THE PROPER NUMBER OF NUMERIC ARGUMENTS, AND SETS A FLAG IF IT DOES
852 - C NOT. IF SPECIAL COMMANDS ARE ENTERED SUBROUTINE PERFORMS FOLLOWING:
853 - C STOP = PROGRAM STOPS
854 - C GEOM = IER IS SET TO 1 SO ALL GEOMETRY PARAMETERS MAY BE RE-ENTERED
855 - C LOAD = IER IS SET TO 2 SO ALL LOAD PARAMETERS MAY BE RE-ENTERED
856 - C
857 - C * SYNTAX * Q(QUEST*72, MARG, ANS*72)
858 - C      QUEST = QUESTION PROMPT PRINTED TO USER
859 - C      MARG = NO. OF NUMERIC ARGUMENTS EXPECTED
860 - C      ANS = ANSWER READ AND PASSED BACK AS CHARACTER STRING
861 - C * CALL BY * WALL
862 - C * CALLS. * WL, CN72
863 - C
864 - C
865 - C SUBROUTINE Q(QUEST, MARG, ANS)
866 - C IMPLICIT REAL(A-H, O-Z)
867 - C CHARACTER QUEST*72, ANS*72
868 - C COMMON /Q1/ IGF2, IGF1, IGF0, IGF2
869 - C CALL WL(QUEST, NQUEST)
870 - C ITRY=0
871 - 10 CONTINUE
872 - IF(QUEST(1:4).NE.' ') WRITE(108, 30) NQUEST, QUEST
873 - 30 FORMAT(1X, AN)
874 - READ '(A72)', ANS
875 - CALL CN72(ANS, IND)
876 - IF(IND.EQ.0) RETURN
877 - IF((ABS(IND).GE.MARG).AND.(IND.NE.-MARG)) THEN
878 - RETURN
879 - ELSE IF(ANS(1:4).EQ.'STOP') THEN
880 - PRINT *, '== RUN ABORTED =='
881 - STOP
882 - ELSE IF(ANS(1:4).EQ.'GEOM') THEN
883 - IF(IGF2.EQ.1) THEN
884 - WRITE(108, 40)
885 - 40 FORMAT(' You are already in geometry edit mode!')
886 - + ' Proceed with normal entry of data:')
887 - GOTO 10
888 - ELSE
889 - END IF
890 - CALL PG(108)
891 - ANS=' '

```



```

892 - RETURN
893 - ELSE IF (ANS(1:4).EQ.'LEGAL') THEN
894 - IF (IG(1).EQ.1) IG(1) =
895 - CALL FCT106
896 - ANS(1) = ' '
897 - RETURN
898 - ELSE
899 - ITH=ITH+1
900 - IF (ITH.GE.3) THEN
901 - PRINT *, 'unable to find answer - returning zero values.'
902 - ANS(1) = ' '
903 - RETURN
904 - ELSE
905 - PRINT *,
906 - 'Incorrect numerical value or too few arguments entered.'
907 - PRINT *, ANS
908 - GOTO 10
909 - END IF
910 - END IF
911 - END
912 - C
913 - C *****
914 - C **** WL **** P. KALAB, VERSION 1.0, JANUARY 15, 1983. ****
915 - C SUBROUTINE IS USED TO FIND THE END OF A QUESTION (DESIGNATED BY
916 - C '?', ':', OR ' ') PASSED IN ARGUMENT 'STRING' AND RETURN
917 - C THE LENGTH IN ARGUMENT 'L'.
918 - C ** SYNTAX ** WL(STRING*72,L)
919 - C * CALL BY * C
920 - C
921 - SUBROUTINE WL(STRING,L)
922 - CHARACTER STRING*72
923 - IMPLICIT INTEGER(1-N)
924 - L=1
925 - 10 IF ((STRING(L:L).EQ.'?') .OR. (STRING(L:L).EQ.':'))
926 - + .OR. (L.EQ.72) .OR. (STRING(L:L+3).EQ.' ')) RETURN
927 - L=L+1
928 - GOTO 10
929 - END
930 - C
931 - C *****
932 - C **** CN72 **** P. KALAB; VERSION 1.2; JANUARY 25, 1983. *****
933 - C SUBROUTINE RECEIVES AN ALPHANUMERIC STRING IN 'WORD' AND PASSES
934 - C BACK THE NUMBER OF ERROR-FREE NUMERIC FIELDS IN 'ABS(IND)'.
935 - C IF ERRORS ARE ENCOUNTERED, 'IND' BECOMES NEGATIVE.
936 - C * SYNTAX * CN72(CHAR*72,IND)
937 - C CHAR = STRING SCANNED FOR NUMERICAL INPUT
938 - C IND = INTEGER INDICATING ERRORS ENCOUNTERED
939 - C * CALL BY * END
940 - C
941 - SUBROUTINE CN72(WORD,IND)
942 - CHARACTER WORD*72,N1*10
943 - DATA N1/'0123456789'/
944 - IER=0
945 - IEL=72
946 - L=0
947 - IND=-1
948 - 10 CONTINUE
949 - IF (IER.EQ.0) THEN
950 - IND=IND+1
951 - ELSE
952 - IF (IND.EQ.0) IND=-1; RETURN
953 - IND=-1*ABS(IND)
954 - GOTO 600
955 - END IF
956 - ISIGN=0
957 - IDOT=0
958 - 20 CONTINUE
959 - L=L+1
960 - IF (WORD(L:L).EQ.' ') THEN

```

```

961 - IF(L,GE,IEL) GOTO 600
962 - GOTO 20
963 - ELSE
964 - IF(L,GE,IEL) GOTO 600
965 - END IF
966 - IF((WORD(L:L),EQ,'+'),OR,(WORD(L:L),EQ,'-')) THEN
967 - ISIGN=1
968 - 30 CONTINUE
969 - L=L+1
970 - IF(L,GE,IEL) GOTO 500
971 - IF(WORD(L:L),EQ,' ') GOTO 30
972 - IF(WORD(L:L),EQ,'.') THEN
973 - IDOT=1
974 - L=L+1
975 - IF(L,GE,IEL) GOTO 500
976 - ELSE
977 - END IF
978 - DO 40 I=1,10
979 - IF(WORD(L:L),EQ,N1(I:1)) GOTO 100
980 - 40 CONTINUE
981 - GOTO 500
982 - ELSE IF(WORD(L:L),EQ,'') THEN
983 - GOTO 10
984 - ELSE IF(WORD(L:L),EQ,'.') THEN
985 - ISIGN=1
986 - IDOT=1
987 - L=L+1
988 - IF(L,GE,IEL) GOTO 500
989 - DO 50 I=1,10
990 - IF(WORD(L:L),EQ,N1(I:1)) GOTO 100
991 - 50 CONTINUE
992 - GOTO 500
993 - ELSE
994 - DO 60 I=1,10
995 - IF(WORD(L:L),EQ,N1(I:1)) GOTO 100
996 - 60 CONTINUE
997 - GOTO 500
998 - END IF
999 - 100 CONTINUE
1000 - L=L+1
1001 - IF(L,GE,IEL) GOTO 600
1002 - DO 110 I=1,10
1003 - IF(WORD(L:L),EQ,N1(I:1)) GOTO 100
1004 - 110 CONTINUE
1005 - IF((WORD(L:L),EQ,''),OR,(WORD(L:L),EQ,' ')) GOTO 10
1006 - IF((WORD(L:L),EQ,'.')AND,(IDOT,EQ,0)) THEN
1007 - IDOT=1
1008 - GOTO 100
1009 - ELSE
1010 - END IF
1011 - IF(WORD(L:L),EQ,'E') THEN
1012 - ISIGN=0
1013 - 120 CONTINUE
1014 - L=L+1
1015 - IF(WORD(L:L),EQ,' ') GOTO 120
1016 - IF(((WORD(L:L),EQ,'+'),OR,(WORD(L:L),EQ,'-'))AND,
1017 - (ISIGN,EQ,0)) THEN
1018 - ISIGN=1
1019 - GOTO 120
1020 - ELSE
1021 - END IF
1022 - DO 130 I=1,10
1023 - IF(WORD(L:L),EQ,N1(I:1)) GOTO 140
1024 - 130 CONTINUE
1025 - GOTO 500
1026 - 140 CONTINUE
1027 - L=L+1
1028 - IF((WORD(L:L),EQ,''),OR,(WORD(L:L),EQ,' ')) GOTO 10
1029 - DO 150 I=1,10

```

```

1030 -      IF (WORD(L:L).EQ.NI(1:1)) GOTO 160
1031 - 150  CONTINUE
1032 -      GOTO 500
1033 - 160  CONTINUE
1034 -      L=L+1
1035 -      IF (L.GE.1EL) GOTO 600
1036 -      IF ((WORD(L:L).EQ.' ') .OR. (WORD(L:L).EQ.' ')) GOTO 10
1037 -      ELSE
1038 -      END IF
1039 - 500  CONTINUE
1040 -      IER=IER+1
1041 -      GOTO 10
1042 - 600  CONTINUE
1043 -      IF (IND.LT.0) IND=IND-1
1044 -      RETURN
1045 -      END
1046 - C
1047 - C *****
1048 - C **** OCHECK **** P. KALAB; VERSION 1.2; SEPTEMBER 19, 1983. ****
1049 - C SUBROUTINE WHICH OPENS FILES AND CHECKS FOR ERROR CONDITIONS. THE
1050 - C USER HAS 3 CHANCES TO CORRECT ANY MISTAKES HE MAY HAVE MADE.
1051 - C * SYNTAX * OCHECK(11,CHAR1*36,CHAR2*7,CHAR3*7,12)
1052 - C * VARIABLES * 11 = DEVICE NUMBER; CHAR1 = FILE NAME; CHAR2 = TYPE;
1053 - C * CHAR3 = ACCESS; 12 = ERROR CODE (0=NONE, 1=ERROR)
1054 - C
1055 -      SUBROUTINE OCHECK(DEV,STRING,TYPE,AX,ER)
1056 -      IMPLICIT REAL(A-H,O-Z)
1057 -      INTEGER DEV,ER
1058 -      CHARACTER STRING*36,TYPE*7,AX*10,RESP*1
1059 -      IREP=0
1060 - 10  CONTINUE
1061 -      OPEN(UNIT=DEV,NAME=STRING,STATUS=TYPE,ACCESS=AX,FORM='FORMATTED',
1062 -      + ERR=100)
1063 -      ER=0
1064 -      RETURN
1065 - 100  CONTINUE
1066 -      IF (TYPE(1:3).EQ.'OLD') THEN
1067 -      PRINT *,
1068 -      + 'ERROR - File is not in directory or has incorrect access.'
1069 -      ELSE IF (TYPE(1:3).EQ.'NEW') THEN
1070 -      PRINT *, 'ERROR - File already exists in directory.'
1071 -      PRINT *, 'Do you wish to overwrite it?'
1072 -      READ '(A1)',RESP
1073 -      IF (RESP.EQ.'Y') THEN
1074 -      OPEN(UNIT=DEV,NAME=STRING,STATUS='OLD',FORM='FORMATTED')
1075 -      CLOSE(UNIT=DEV,STATUS='DELETE')
1076 -      OPEN(UNIT=DEV,NAME=STRING,STATUS=TYPE,ACCESS=AX,
1077 -      + FORM='FORMATTED',ERR=100)
1078 -      RETURN
1079 -      ELSE
1080 -      END IF
1081 -      ELSE
1082 -      PRINT *, 'ERROR - Unable to open file.'
1083 -      END IF
1084 -      IREP=IREP+1
1085 -      IF (IREP.GT.3) THEN
1086 -      PRINT *, 'Sorry, unable to open file specified.'
1087 -      ER=1
1088 -      RETURN
1089 -      ELSE
1090 -      PRINT *, 'Enter correct file name:'
1091 -      READ '(A36)',STRING
1092 -      GOTO 10
1093 -      END IF
1094 -      END
1095 - C
1096 - C *****
1097 - C **** ASPECT **** P. KALAB; VERSION 1.0; NOVEMBER 10, 1983. ****
1098 - C SUBROUTINE WHICH CHECKS THE ASPECT RATIO OF THE CRITICAL ELEMENTS

```

```

1099 - C AND PRINTS WARNING MESSAGES IF THE ASPECT RATIO IS EXCEEDED, THE
1100 - C MAXIMUM ASPECT RATIO IS SET AT 10.0, BUT MAY BE ALTERED BY CHANGING
1101 - C THE VALUE TO WHICH VARIABLE "AR" IS SET.
1102 - C * SYNTAX * ASPECT(W1,W2,AR,NEH,IER)
1103 - C      W1 = WIDTH OF WALL 1 (DOUBLE PRECISION)
1104 - C      W2 = WIDTH OF WALL 2 (DOUBLE PRECISION)
1105 - C      AR = MAXIMUM ALLOWABLE ASPECT RATIO
1106 - C      NEH = NO. OF HORIZONTAL ELEMENTS IN WALL 1 (INTEGER)
1107 - C      ALL SUBSEQUENT CALCULATIONS ARE BASED ON THIS VALUE
1108 - C      IER = ERROR INDICATOR SET TO 0 IF ALL ASPECT RATIOS ARE OK
1109 - C * CALL BY * WALL
1110 - C * COMMON * GEOM3(WALL), AR1(GEOM,LOAD)
1111 - C
1112 - C      SUBROUTINE ASPECT(W1,W2,AR,NEH,IGEOM,IER)
1113 - C      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
1114 - C      IMPLICIT INTEGER(I-N)
1115 - C      CHARACTER RESP*1
1116 - C      COMMON /AR1/ NEH1,NEH2,NEV
1117 - C      COMMON /GEOM3/ WMAX,WMIN,BMAX,BMIN
1118 - C
1119 - C      INITIALIZE VARIABLES
1120 - C
1121 - C      WRITE(108,10)
1122 - C      10 FORMAT(' Generating mesh. Please wait ...')
1123 - C      IPASS=0
1124 - C      RMAX=0.0
1125 - C      IER=0
1126 - C      NEH1=NEH
1127 - C      HINC1=W1/DFLOAT(NEH1)
1128 - C      NEV=ANINT((WMAX+WMIN)/(2.0*HINC1))
1129 - C      IF(NEV.LT.1) NEV=1
1130 - C      NEH2=ANINT(W2/HINC1)
1131 - C      IF(NEH2.LT.1) NEH2=1
1132 - C      ICOUNT=0
1133 - C
1134 - C      THIN BEAM, WIDE WALL 1
1135 - C
1136 - C      20 CONTINUE
1137 - C      HINC1=W1/DFLOAT(NEH1)
1138 - C      R=HINC1/(BMIN/2.0)
1139 - C      IF(R.GT.AR) THEN
1140 - C          ICOUNT=ICOUNT+1
1141 - C          IF(ICOUNT.GT.100) THEN
1142 - C              WRITE(108,30) R,NEH1
1143 - C              30 FORMAT(' Ratio HINC1/(BMIN/2) =',F6.1,' NEH1 =',14)
1144 - C              ICOUNT=0
1145 - C              IER=1
1146 - C              GOTO 50
1147 - C          ELSE
1148 - C              NEH1=NEH1+1
1149 - C              GOTO 20
1150 - C          END IF
1151 - C      ELSE
1152 - C          IF(R.GT.RMAX) RMAX=R
1153 - C          ICOUNT=0
1154 - C      END IF
1155 - C
1156 - C      THIN BEAM, WIDE WALL 2
1157 - C
1158 - C      50 CONTINUE
1159 - C      HINC2=W2/DFLOAT(NEH2)
1160 - C      R=HINC2/(BMIN/2.0)
1161 - C      IF(R.GT.AR) THEN
1162 - C          ICOUNT=ICOUNT+1
1163 - C          IF(ICOUNT.GT.100) THEN
1164 - C              WRITE(108,60) R,NEH2
1165 - C              60 FORMAT(' Ratio HINC2/(BMIN/2) =',F6.1,' NEH2 =',14)
1166 - C              ICOUNT=0
1167 - C          ELSE
1168 - C              NEH2=NEH2+1
1169 - C              GOTO 50
1170 - C          END IF
1171 - C      ELSE
1172 - C          IF(R.GT.RMAX) RMAX=R
1173 - C          ICOUNT=0
1174 - C      END IF
1175 - C
1176 - C      IER=2

```

```

1168 -      GOTO 80
1169 -      ELSE
1170 -      NEH2=NEH2+1
1171 -      GOTO 50
1172 -      END IF
1173 -      ELSE
1174 -      IF (R,GT,RMAX) RMAX=R
1175 -      ICOUNT=0
1176 -      END IF
1177 - C
1178 - C LOW AND WIDE WALL 1
1179 - C
1180 - 80 CONTINUE
1181 - HINC1=1/DFLOAT(NEH1)
1182 - VINC=VMIN/DFLOAT(NEV)
1183 - R=HINC1/VINC
1184 - IF (R,GT,AR) THEN
1185 -   ICOUNT=ICOUNT+1
1186 -   IF (ICOUNT,GT,100) THEN
1187 -     WRITE(108,90) R,NEH1
1188 - 90   FORMAT(' Ratio HINC1/VMIN =',F6.1,' NEH1 =', 14)
1189 -     ICOUNT=0
1190 -     IER=3
1191 -     GOTO 110
1192 -   ELSE
1193 -     NEH1=NEH1+1
1194 -     GOTO 80
1195 -   END IF
1196 - ELSE
1197 - IF (R,GT,RMAX) RMAX=R
1198 - ICOUNT=0
1199 - END IF
1200 - C
1201 - C LOW AND WIDE WALL 2
1202 - C
1203 - 110 CONTINUE
1204 - HINC2=1/DFLOAT(NEH2)
1205 - VINC=VMIN/DFLOAT(NEV)
1206 - R=HINC2/VINC
1207 - IF (R,GT,AR) THEN
1208 -   ICOUNT=ICOUNT+1
1209 -   IF (ICOUNT,GT,100) THEN
1210 -     WRITE(108,120) R,NEH2
1211 - 120   FORMAT(' Ratio HINC2/VMIN =',F6.1,' NEH2 =', 14)
1212 -     ICOUNT=0
1213 -     IER=4
1214 -     GOTO 140
1215 -   ELSE
1216 -     NEH2=NEH2+1
1217 -     GOTO 110
1218 -   END IF
1219 - ELSE
1220 - IF (R,GT,RMAX) RMAX=R
1221 - ICOUNT=0
1222 - END IF
1223 - C
1224 - C HIGH AND NARROW WALL 1
1225 - C
1226 - 140 CONTINUE
1227 - HINC1=1/DFLOAT(NEH1)
1228 - VINC=VMAX/DFLOAT(NEV)
1229 - R=VINC/HINC1
1230 - IF (R,GT,AR) THEN
1231 -   ICOUNT=ICOUNT+1
1232 -   IF (ICOUNT,GT,100) THEN
1233 -     WRITE(108,150) R,NEV
1234 - 150   FORMAT(' Ratio VMAX/HINC1 =',F6.1,' NEV =', 14)
1235 -     ICOUNT=0
1236 -     IER=5

```

```

1237 -      GOTO 170
1238 -      ELSE
1239 -      NEV=NEV+1
1240 -      GOTO 140
1241 -      END IF
1242 -      ELSE
1243 -      IF(R.GT,RMAX) RMAX=R
1244 -      ICOUNT=0
1245 -      END IF
1246 - C
1247 - C HIGH AND NARROW WALL 2
1248 - C
1249 - 170 CONTINUE
1250 -      HINC2=W2/DFLOAT(NEH2)
1251 -      VINC=WMAX/DFLOAT(NEV)
1252 -      R=VINC/HINC2
1253 -      IF(R.GT,AR) THEN
1254 -      ICOUNT=ICOUNT+1
1255 -      IF(ICOUNT.GT,100) THEN
1256 -      WRITE(108,180) R,NEV
1257 - 180      FORMAT(' Ratio WMAX/HINC2 =',F6.1,' NEV =',14)
1258 -      ICOUNT=0
1259 -      IER=6
1260 -      GOTO 200
1261 -      ELSE
1262 -      NEV=NEV+1
1263 -      GOTO 170
1264 -      END IF
1265 -      ELSE
1266 -      IF(R.GT,RMAX) RMAX=R
1267 -      ICOUNT=0
1268 -      END IF
1269 - C
1270 - C HIGH BEAM IN WALL 1
1271 - C
1272 - 200 CONTINUE
1273 -      HINC1=W1/DFLOAT(NEH1)
1274 -      R=(BMAX/2.00)/HINC1
1275 -      IF(R.GT,AR) THEN
1276 -      IER=7
1277 -      WRITE(108,210) R,BMAX,NEH1
1278 - 210      FORMAT(' Desired geometry exceeds limits of this program.'/
1279 -      + ' Aspect ratio =',F6.1,' Beam height =',F9.1,
1280 -      + ' No. of hor. elem. in Wall 1 =',16/
1281 -      + ' Either beam is too deep or mesh is too fine.')
1282 -      ELSE
1283 -      IF(R.GT,RMAX) RMAX=R
1284 -      END IF
1285 - C
1286 - C HIGH BEAM IN WALL 2
1287 - C
1288 -      HINC2=W2/DFLOAT(NEH2)
1289 -      R=(BMAX/2.00)/HINC2
1290 -      IF(R.GT,AR) THEN
1291 -      IER=8
1292 -      WRITE(108,230) R,BMAX,NEH2
1293 - 230      FORMAT(' Desired geometry exceeds limits of this program.'/
1294 -      + ' Aspect ratio =',F6.1,' Beam height =',F9.1,
1295 -      + ' No. of hor. elem. in Wall 2 =',16/
1296 -      + ' Either beam is too deep or mesh is too fine.')
1297 -      ELSE
1298 -      IF(R.GT,RMAX) RMAX=R
1299 -      END IF
1300 - C
1301 - C MAKE SECOND PASS TO MAKE SURE ALL ASPECT RATIOS ARE STILL OK
1302 - C
1303 -      IF(IPASS.EQ,0) THEN
1304 -      IER=0
1305 -      IPASS=1

```

```

1306 -      GOTO 20
1307 -      ELSE
1308 -      END IF
1309 - 28. CONTINUE
1310 -      IF (IER.GT.0) THEN
1311 -      WRITE(106,300)
1312 - 30. FORMAT('
1313 -      + ' WARNING - Mesh which will be generated may produce incorrect'
1314 -      + ' results. Do you wish to proceed (Y/N)?')
1315 -      READ '(A1)',RESP
1316 -      IF (RESP.EQ.'Y') THEN
1317 -      IER=0
1318 -      ELSE IF (RESP.EQ.'N') THEN
1319 -      RETURN
1320 -      ELSE
1321 -      PRINT *, 'Please answer "Y" or "N" '
1322 -      GOTO 28.
1323 -      END IF
1324 -      ELSE
1325 -      END IF
1326 -      WRITE(106,310) RMAX,AR,NEH1,NEH2,NEV+2
1327 -      IF (IGEOM.EQ.0) WRITE(2,310) RMAX,AR,NEH1,NEH2,NEV+2
1328 - 310 FORMAT('/// Highest aspect ratio =',F6.1,' ---
1329 -      + ' Max. allowable ratio =',F6.1/
1330 -      + ' No. of elements in horizontal direction in Wall 1 =',15/
1331 -      + ' No. of elements in horizontal direction in Wall 2 =',15/
1332 -      + ' No. of elements in vertical direction per storey =',15//)
1333 -      WRITE(106,320)
1334 - 320 FORMAT('
1335 -      + ' If satisfied press RETURN, otherwise enter any character:')
1336 -      READ '(A1)',RESP
1337 -      IF ((RESP.EQ.'Y').OR.(RESP.EQ.' ')) THEN
1338 -      ELSE IF (RESP.EQ.'S') THEN
1339 -      PRINT *, '== RUN ABORTED =='
1340 -      STOP
1341 -      ELSE
1342 -      IER=-1
1343 -      END IF
1344 -      RETURN
1345 -      END
1346 - C
1347 - C *****
1348 - C **** GEOM **** P. KALAB; VERSION 1.0; NOVEMBER 11, 1983. ****
1349 - C SUBROUTINE WHICH WRITES ALL OF THE GEOMETRY CARDS REQUIRED FOR A
1350 - C STARDYNE RUN.
1351 - C * SYNTAX * GEOM(DEN,E,W1,GAP,W2,NFL,IEL)
1352 - C      DEN = CONCRETE MASS DENSITY (kg/cu.m)
1353 - C      E = MODULUS OF ELASTICITY
1354 - C      W1 = LENGTH OF WALL 1
1355 - C      GAP = DISTANCE BETWEEN TWO WALLS
1356 - C      W2 = LENGTH OF WALL 2
1357 - C      NFL = NUMBER OF STOREYS
1358 - C      L = NO. OF QUADRILATERAL ELEMENTS
1359 - C * CALL BY * WALL
1360 - C * COMMON * AR1(ASPECT), GEOM2(WALL,PG,PL), REST1(WALL), FILES(WALL)
1361 - C
1362 - C ** IMPORTANT VARIABLES **
1363 - C NODE = NO. OF NODES IN MODEL
1364 - C IB = TOTAL NO. OF BEAMS (NFL ELASTIC BEAMS AND (IB-NFL) RIGID)
1365 - C IEL = NO. OF QUADRILATERAL ELEMENTS
1366 - C NW2 = STARTING NODE FOR WALL 2
1367 - C NEV = NO. OF VERTICAL ELEMENTS PER STOREY (WITHOUT TWO ROWS FOR
1368 - C      BEAM)
1369 - C NEH1 = NO. OF HORIZONTAL ELEMENTS IN WALL 1
1370 - C NEH2 = NO. OF HORIZONTAL ELEMENTS IN WALL 2
1371 - C PT(1)= WALL THICKNESS AT STOREY 1
1372 - C WH(1)= WALL HEIGHT AT STOREY 1
1373 - C BH(1)= BEAM HEIGHT AT STOREY 1
1374 - C X,Y,Z NODAL COORDINATES

```

```

1375 - C JA, JE, JC, JE = ELEMENT CORNER NODES
1376 - C
1377 - C OTHER VARIABLES ARE AS DEFINED IN STADYNE MANUAL, AS HAVE BEEN
1378 - C DEFINED ELSEWHERE IN THE PROGRAM, OR ARE ONLY TEMPORARY COUNTERS
1379 - C AND TEMPORARY VARIABLES.
1380 - C
1381 - C
1382 - C SUBROUTINE GEOM(DEN,E,W1,GAP,W2,NFL,IEC)
1383 - C IMPLICIT DOUBLE PRECISION(A-H,O-Z)
1384 - C IMPLICIT INTEGER(I-N)
1385 - C CHARACTER RESP*1,COM*4,SFILE*36,CFILE*36
1386 - C DIMENSION PT(50),WH(50),BH(50)
1387 - C COMMON /AR1/ NEH1,NEH2,NEV
1388 - C COMMON /GEOM2/ PT,WH,BH
1389 - C COMMON /REST1/ NODE,MW2,IGEOM,HEF
1390 - C COMMON /FILES/ SFILE,CFILE
1391 - C
1392 - C WRITE START CARD
1393 - C
1394 - C WRITE(108,30)
1395 - C 30 FORMAT(' Processing geometry data. Please wait ...')
1396 - C NFLE=0
1397 - C G=9.8100
1398 - C JDOF=6
1399 - C IRENUM=0
1400 - C IT4=0
1401 - C WRITE(1,40) NFLE,G,JDOF,IRENUM,IT4
1402 - C 40 FORMAT('START',3X,14,F10.2,2I5,6X,12,9'2',13')
1403 - C
1404 - C PLOT CARD MAY BE INSERTED HERE, BUT NO PROVISIONS HAVE BEEN MADE
1405 - C IN THIS PREPROCESSOR FOR PLOTTING
1406 - C
1407 - C STRUCTURE DESCRIPTION CARD
1408 - C
1409 - C WRITE(1,50) NFL,W1,GAP,W2
1410 - C 50 FORMAT(1X,13,'-floor shear walls, Wall 1:',F6.1,' m, Gap:',F6.1,
1411 - C + ' m; Wall 2:',F6.1,' m')
1412 - C
1413 - C MATLG CARD - DENSITY IS CONVERTED FROM MASS DENSITY TO WEIGHT DENS.
1414 - C UNITS ARE KN AND METRES.
1415 - C
1416 - C PR=0.1500
1417 - C DEN=DEN*9.810-3
1418 - C ALPHA=0.00
1419 - C CHDAMP=0.00
1420 - C E=E*1000.00
1421 - C WRITE(1,60) E,PR,DEN,ALPHA,CHDAMP
1422 - C 60 FORMAT('MATLG 1 CONCRETE ',5E10.5)
1423 - C WRITE(1,300)
1424 - C
1425 - C NODAL COORDINATE TABLE FOR WALL 1
1426 - C
1427 - C NODE=0
1428 - C X=0.00
1429 - C Y=0.00
1430 - C HINC=W1/DFLOAT(NEH1)
1431 - C DO 100 I=1,NEH1+1
1432 - C NODE=NODE+1
1433 - C WRITE(1,70) NODE,X,Y
1434 - C 70 FORMAT('NODE',14,2F16.4)
1435 - C X=X+HINC
1436 - C 100 CONTINUE
1437 - C DO 110 I=1,NFL
1438 - C WH(I)=WH(I)-BH(I)
1439 - C 110 CONTINUE
1440 - C DO 180 I=1,NFL
1441 - C VINC=WH(I)/DFLOAT(NEV)
1442 - C BINC=BH(I)/2.00
1443 - C DO 140 J=1,NEV

```



```

1444 -      X=0.0
1445 -      Y=Y+VINC
1446 -      DO 120 K=1,NEH1+1
1447 -        NODE=NODE+1
1448 -        WRITE(1,70) NODE,X,Y
1449 -        X=X+HINC
1450 - 120 CONTINUE
1451 - 140 CONTINUE
1452 -      DO 160 J=1,2
1453 -        X=0.0
1454 -        Y=Y+BINC
1455 -        DO 150 K=1,NEH1+1
1456 -          NODE=NODE+1
1457 -          WRITE(1,70) NODE,X,Y
1458 -          X=X+HINC
1459 - 150 CONTINUE
1460 - 160 CONTINUE
1461 - 180 CONTINUE
1462 - C
1463 - C NODAL COORDINATE TABLE FOR WALL 2
1464 - C
1465 -      NW2=NODE+1
1466 -      X=W1+GAP
1467 -      Y=0.0
1468 -      HINC=W2/DFLOAT(NEH2)
1469 -      DO 200 I=1,NEH2+1
1470 -        NODE=NODE+1
1471 -        WRITE(1,70) NODE,X,Y
1472 -        X=X+HINC
1473 - 200 CONTINUE
1474 -      DO 280 I=1,NFL
1475 -        VINC=W1(1)/DFLOAT(NEV)
1476 -        BINC=BH(1)/2.0
1477 -        DO 240 J=1,NEV
1478 -          X=W1+GAP
1479 -          Y=Y+VINC
1480 -          DO 220 K=1,NEH2+1
1481 -            NODE=NODE+1
1482 -            WRITE(1,70) NODE,X,Y
1483 -            X=X+HINC
1484 - 220 CONTINUE
1485 - 240 CONTINUE
1486 -          DO 260 J=1,2
1487 -            X=W1+GAP
1488 -            Y=Y+BINC
1489 -            DO 250 K=1,NEH2+1
1490 -              NODE=NODE+1
1491 -              WRITE(1,70) NODE,X,Y
1492 -              X=X+HINC
1493 - 250 CONTINUE
1494 - 260 CONTINUE
1495 - 280 CONTINUE
1496 -      WRITE(1,300)
1497 -      FORMAT('END')
1498 - C
1499 - C WALL IS RESTRAINED AT BASE SO THAT THERE IS NO MOVEMENT.
1500 - C OTHERWISE STRUCTURE IS TREATED AS PLANAR UNDER DEFINITION OF
1501 - C QUADPLATE PROPERTIES.
1502 - C
1503 -      NS=1
1504 -      NE=NEH1+1
1505 -      INC=1
1506 -      WRITE(1,310) NS,NE,INC,1,2,3,4,5,6
1507 - 310 FORMAT('RESTG ',314,611)
1508 -      NS=NW2
1509 -      NE=NEH2+NW2
1510 -      WRITE(1,310) NS,NE,INC,1,2,3,4,5,6
1511 -      WRITE(1,300)
1512 - C

```

```

1513 - C ELASTIC BEAM TABLE
1514 - C
1515 -     JA=1+NEH1+(NEH1+1)*(NEV+1)
1516 -     JB=NM2+(NEH2+1)*(NEV+1)
1517 -     IJA=(NEH1+1)*(NEV+2)
1518 -     IJB=(NEH2+1)*(NEV+2)
1519 -     JC=JA+NEH1+1
1520 -     IJC=NEH1+1
1521 -     IB=1
1522 -     DO 370 I=1,NFL
1523 -         WRITE(1,360) IB,JA,9999,JB,0,JC,1,IB
1524 - 360     FORMAT('BRECT',3X,14,4X,714)
1525 -         JA=JA+1
1526 -         JB=JB+1
1527 -         JC=JA+1
1528 -         IB=IB+1
1529 - 370     CONTINUE
1530 - C
1531 - C RIGID BEAM TABLE
1532 - C
1533 -     JA=(NEH1+1)*(NEV+1)
1534 -     IJA=JA
1535 -     JC2=JA
1536 -     IJC2=JA
1537 -     IPROP=IB
1538 -     DO 390 I=1,NFL
1539 -         JB=JA+1+NEH1
1540 -         JC=JA-1
1541 -         WRITE(1,360) IB,JA,9999,JB,0,JC,1,IPROP
1542 -         JA=JA+NEH1+1
1543 -         JB=JA+1+NEH1
1544 -         JC=JA-1
1545 -         IB=IB+1
1546 -         WRITE(1,360) IB,JA,9999,JB,0,JC,1,IPROP
1547 -         JA=JA+1
1548 -         IB=IB+1
1549 - 390     CONTINUE
1550 -     JA=NM2+NEV*(NEH2+1)
1551 -     IJA=(NEV+1)*(NEH2+1)
1552 -     DO 400 I=1,NFL
1553 -         JB=JA+1+NEH2
1554 -         WRITE(1,360) IB,JA,9999,JB,0,JC2,1,IPROP
1555 -         JA=JA+NEH2+1
1556 -         JB=JA+1+NEH2
1557 -         JC2=JC2+NEH1+1
1558 -         IB=IB+1
1559 -         WRITE(1,360) IB,JA,9999,JB,0,JC2,1,IPROP
1560 -         JA=JA+1
1561 -         JC2=JC2+1
1562 -         IB=IB+1
1563 - 400     CONTINUE
1564 -     WRITE(1,300)
1565 - C
1566 - C BEAM PROPERTY TABLE FOR ELASTIC BEAMS. CTORS IS A TORSIONAL
1567 - C CONSTANT WHICH IS CALCULATED (BY DEFAULT) BY STARDYNE.
1568 - C
1569 -     SSF2=1.5D0
1570 -     SSF3=1.5D0
1571 -     WGH1=0.0D0
1572 -     DO 450 I=1,NFL
1573 -         WRITE(1,420) I,PT(I),BH(I),SSF2,SSF3,WGH1
1574 - 420     FORMAT('BPROP2',2X,14,2F10.2,10X,3F10.2)
1575 -         K=1
1576 - 450     CONTINUE
1577 -     WRITE(1,460) K+1,1
1578 - 460     FORMAT('BPROP6',2X,14,110)
1579 -     WRITE(1,300)
1580 - C
1581 - C QUADRILATERAL PLATE ELEMENTS

```

22
OF / DE

```

1582 - C
1583 - IEL=1
1584 - ITYPE=2
1585 - MAT=1
1586 - IORD=2
1587 - AXIS=0.00
1588 - JA=0
1589 - DO 560 I=1,NFL
1590 - DO 550 J=1,NEV+2
1591 - JA=JA+1
1592 - DO 540 K=1,NEH1
1593 - JB=JA+1
1594 - JD=JB+NEH1
1595 - JC=JD+1
1596 - WRITE(1,530) IEL,JA,JB,JC,JD,MAT,ITYPE,PT(1),
1597 - + AXIS,IORD
1598 - 530 FORMAT('QUADE',17,518,14,F8.3,F6.1,12)
1599 - IEL=IEL+1
1600 - JA=JA+1
1601 - 540 CONTINUE
1602 - 550 CONTINUE
1603 - 560 CONTINUE
1604 - JA=JA+2-1
1605 - DO 600 I=1,NFL
1606 - DO 590 J=1,NEV+2
1607 - JA=JA+1
1608 - DO 580 K=1,NEH2
1609 - JB=JA+1
1610 - JD=JB+NEH2
1611 - JC=JD+1
1612 - WRITE(1,530) IEL,JA,JB,JC,JD,MAT,ITYPE,PT(1),
1613 - + AXIS,IORD
1614 - IEL=IEL+1
1615 - JA=JA+1
1616 - 580 CONTINUE
1617 - 590 CONTINUE
1618 - 600 CONTINUE
1619 - WRITE(1,300)
1620 - WRITE(1,610)
1621 - 610 FORMAT('ENDGEOM')
1622 - C
1623 - C PRINT SUMMARY OF PARAMETERS ENTERED FOR USER AND GIVE HIM THE
1624 - C OPTION OF ABORTING OR CHANGING GEOMETRY.
1625 - C
1626 - IEL=IEL-1
1627 - WRITE(108,650) NODE,NFL,IB-NFL-1,IEL
1628 - 650 FORMAT(/
1629 - + ' No. of nodes: ',15/
1630 - + ' No. of elastic beams: ',15/
1631 - + ' No. of rigid beams: ',15/
1632 - + ' No. of quad. plates: ',15/)
1633 - IF(IGEQM.EQ.0) WRITE(2,650) NODE,NFL,IB-NFL-1,IEL
1634 - WRITE(108,655)
1635 - 655 FORMAT(
1636 - + ' If satisfied press RETURN, otherwise enter any character:')
1637 - READ '(A1)',RESP
1638 - IF(RESP.NE.' ') THEN
1639 - 660 CONTINUE
1640 - PRINT *
1641 - + 'Enter "STOP" to abort, "GEOM" to edit geometry, or "GO" to',
1642 - + ' continue:'
1643 - READ '(A4)',COM
1644 - IF(COM.EQ.'STOP') THEN
1645 - CLOSE(UNIT=1,STATUS='DELETE')
1646 - CLOSE(UNIT=2,STATUS='DELETE')
1647 - STOP
1648 - ELSE IF(COM.EQ.'GEOM') THEN
1649 - DEN=DEN/9.81D-3
1650 - E=E/1000.00

```

```

1651 - DO 670 I=1,NFL
1652 -   WH(I)=WH(I)+BH(I)
1653 - 67C CONTINUE
1654 -   IER=1
1655 -   RETURN
1656 -   ELSE IF (COM(1:2).EQ.'GO') THEN
1657 -     GOTO 680
1658 -   ELSE
1659 -     PRINT *, 'Unrecognizable command: ', COM
1660 -     GOTO 660
1661 -   END IF
1662 -   ELSE
1663 -   END IF
1664 - 68C CONTINUE
1665 -   IF (IGEOM.EQ.1) THEN
1666 -     WRITE(108,690) SFIL
1667 - 69C FORMAT(/
1668 -   + ' Geometry data has been written on file ', A36//
1669 -   + ' ++++ NORMAL TERMINATION ++++' )
1670 -   CLOSE(UNIT=1, STATUS='KEEP')
1671 -   IER=0
1672 -   RETURN
1673 -   ELSE
1674 -     WRITE(108,700)
1675 - 700 FORMAT(/ ' Processing static loads. Please wait ...' )
1676 -     IER=0
1677 -     RETURN
1678 -   END IF
1679 -   END
1680 - C
1681 - C *****
1682 - C **** LOAD **** P. KALAB; VERSION 1.0; NOVEMBER 12, 1983. ****
1683 - C SUBROUTINE TO CALCULATE THE BEAM AND NODAL LOADS FROM THE PARAMETERS
1684 - C ENTERED BY THE USER IN "WALL-MAIN". THE LOADS ARE:
1685 - C 1. GRAVITY LOADS APPLIED AS NODAL LOADS
1686 - C 2. UDL LOAD ON EACH FLOOR APPLIED TO NODES AT THE TOP OF EACH FLOOR
1687 - C   AND AS A UDL ON THE CONNECTING BEAMS
1688 - C 3. LINEARLY DISTRIBUTED LATERAL LOAD ACTING ON WALL 1, APPLIED AS
1689 - C   NODAL LOADS
1690 - C 4. CONCENTRATED LATERAL LOADS APPLIED AT THE TOP OF EACH FLOOR AS
1691 - C   NODAL LOADS.
1692 - C
1693 - C * SYNTAX * LOAD(DEN,GAP,NFL,NEND,NN2,IEL)
1694 - C   DEN = CONCRETE WEIGHT DENSITY (N/cu.m) UNIFORM FOR
1695 - C   STRUCTURE
1696 - C   GAP = LENGTH OF BEAM
1697 - C   NFL = NO. OF FLOORS
1698 - C   NEND = TOTAL NO. OF NODES IN STRUCTURE
1699 - C   NN2 = STARTING NODE OF WALL 2
1700 - C   IEL = NO. OF QUADRILATERAL ELEMENTS
1701 - C
1702 - C * CALL BY * WALL
1703 - C * COMMON * LOAD1(WALL,PL), AR1(ASPECT,GEOM), GEOM2(WALL,PG)
1704 - C   FILES(WALL,GEOM)
1705 - C
1706 - C ** IMPORTANT VARIABLES ** AS DEFINED IN OTHER SUBROUTINES
1707 - C
1708 - C SUBROUTINE LOAD(NFL,NEND,NN2,IEL)
1709 - C IMPLICIT DOUBLE PRECISION(A-H,O-Z)
1710 - C IMPLICIT INTEGER(I-N)
1711 - C CHARACTER SFIL*36,CFIL*36
1712 - C DIMENSION PT(50),WH(50),BH(50),UDL(50),PLOAD(50)
1713 - C COMMON /GEOM1/ DEN,STR,E,W1,GAP,W2,AR,NEH
1714 - C COMMON /GEOM2/ PT,WH,BH
1715 - C COMMON /LOAD1/ UDL,BLOAD,TLOAD,PLOAD
1716 - C COMMON /FILES/ SFIL,CFIL
1717 - C COMMON /AR1/ NEH1,NEH2,NEV
1718 - C NN2=NN2
1719 - C NODE=NEND

```

```

1720 - C
1721 - C WRITE STATIC DESCRIPTION CARD, RUN CONTROL CARD, AND LOAD CASE
1722 - C DESCRIPTION CARD. INITIALIZE SCRATCH FILE 3 WHERE VERTICAL NODAL
1723 - C LOADS ARE TEMPORARILY STORED AND SCRATCH FILE 4 WHERE LATERAL NODAL
1724 - C LOADS ARE STORED.
1725 - C
1726 -     WRITE(1,50)
1727 -     50 FORMAT(' Static analysis with the following loads:')
1728 -     NOUT=0
1729 -     ITAPE=0
1730 -     ISAVE=0
1731 -     IEROPT=1
1732 -     WRITE(1,60) NOUT,ITAPE,ISAVE,IEROPT
1733 -     60 FORMAT('STATIC',16,4X,214,4X,14)
1734 -     WRITE(1,70)
1735 -     70 FORMAT(' Gravity, Vertical UDL on floors, Lateral LDL & point',
1736 -     * ' loads on Wall 1')
1737 -     OPEN(UNIT=3,NAME='F3',STATUS='UNKNOWN',ACCESS='DIRECT',
1738 -     + FORM='FORMATTED')
1739 -     OPEN(UNIT=4,NAME='F4',STATUS='UNKNOWN',ACCESS='DIRECT',
1740 -     + FORM='FORMATTED')
1741 -     DO 100 I=1,NODE
1742 -         WRITE(3,90,REC=I) 0.00
1743 -     90 FORMAT(1G)
1744 -     100 CONTINUE
1745 -     DO 120 I=1,NW2
1746 -         WRITE(4,90,REC=I) 0.00
1747 -     120 CONTINUE
1748 - C
1749 - C WRITE UDL ACTING ON EACH BEAM. AXIS 1 IS FROM WALL 1 TO WALL 2,
1750 - C AXIS 2 IS UP, AND AXIS 3 IS UP OUT OF PLANE.
1751 - C
1752 -     DO 150 I=1,NFL
1753 -         WRITE(1,140) I,1,0,0.00,4UDL(I),0.00
1754 -     140 FORMAT('BMLoad',16,214,'GL',3F10.2)
1755 -     150 CONTINUE
1756 -     WRITE(1,200)
1757 -     200 FORMAT('END')
1758 - C
1759 - C WRITE GRAVITY VERTICAL LOAD FOR WALL 1
1760 - C
1761 -     NH=NEH1+1
1762 -     NODE=1+NH
1763 -     NV=NEV+1
1764 -     DO 300 I=1,NFL
1765 -         NODE=NODE-NH
1766 -         WIDTH=W1/DFLOAT(NH1)
1767 -         H1=WH(I)/DFLOAT(NEV)
1768 -         H2=BH(I)/2.00
1769 -         WW1=DEN*WIDTH*H1*PT(1)
1770 -         WW2=DEN*WIDTH*H2*PT(1)
1771 -         DO 250 J=1,NV
1772 -             IF((J.EQ.1).OR.(J.EQ.NV)) THEN
1773 -                 F1=0.500
1774 -             ELSE
1775 -                 F1=1.000
1776 -             END IF
1777 -             DO 240 K=1,NH
1778 -                 IF((K.EQ.1).OR.(K.EQ.NH)) THEN
1779 -                     F2=0.500
1780 -                 ELSE
1781 -                     F2=1.000
1782 -                 END IF
1783 -                 WGT=READ(3,90,REC=NODE) WGT
1784 -                 WGT=WGT+WW1*F1*F2
1785 -                 WRITE(3,90,REC=NODE) WGT
1786 -                 NODE=NODE+1
1787 -     240 CONTINUE
1788 -     250 CONTINUE

```

```

1789 -      NODE=NODE-NH
1790 -      DO 270 J=1,3
1791 -        IF(J.EQ.2) THEN
1792 -          F1=1.000
1793 -        ELSE
1794 -          F1=0.500
1795 -        END IF
1796 -      DO 260 K=1,NH
1797 -        IF((K.EQ.1).OR.(K.EQ.NH)) THEN
1798 -          F2=0.500
1799 -        ELSE
1800 -          F2=1.000
1801 -        END IF
1802 -      READ(3,90,REC=NODE) WGT
1803 -      WGT=WGT+WW2*F1*F2
1804 -      WRITE(3,90,REC=NODE) WGT
1805 -      NODE=NODE+1
1806 - 260 CONTINUE
1807 - 270 CONTINUE
1808 - 300 CONTINUE
1809 - C
1810 - C GRAVITY LOAD FOR WALL 2
1811 - C
1812 -      NH=NEH2+1
1813 -      NODE=NODE+NH
1814 -      DO 400 I=1,NFL
1815 -        NODE=NODE-NH
1816 -        WIDTH=W2/DFLOAT(NEH2)
1817 -        H1=NH(1)/DFLOAT(NEV)
1818 -        H2=NH(1)/2.DO
1819 -        WW1=DEN*WIDTH*H1*PT(1)
1820 -        WW2=DEN*WIDTH*H2*PT(1)
1821 -      DO 350 J=1,NV
1822 -        IF((J.EQ.1).OR.(J.EQ.NV)) THEN
1823 -          F1=0.500
1824 -        ELSE
1825 -          F1=1.000
1826 -        END IF
1827 -      DO 340 K=1,NH
1828 -        IF((K.EQ.1).OR.(K.EQ.NH)) THEN
1829 -          F2=0.500
1830 -        ELSE
1831 -          F2=1.000
1832 -        END IF
1833 -      READ(3,90,REC=NODE) WGT
1834 -      WGT=WGT+WW1*F1*F2
1835 -      WRITE(3,90,REC=NODE) WGT
1836 -      NODE=NODE+1
1837 - 340 CONTINUE
1838 - 350 CONTINUE
1839 -      NODE=NODE-NH
1840 -      DO 370 J=1,3
1841 -        IF(J.EQ.2) THEN
1842 -          F1=1.000
1843 -        ELSE
1844 -          F1=0.500
1845 -        END IF
1846 -      DO 360 K=1,NH
1847 -        IF((K.EQ.1).OR.(K.EQ.NH)) THEN
1848 -          F2=0.500
1849 -        ELSE
1850 -          F2=1.000
1851 -        END IF
1852 -      READ(3,90,REC=NODE) WGT
1853 -      WGT=WGT+WW2*F1*F2
1854 -      WRITE(3,90,REC=NODE) WGT
1855 -      NODE=NODE+1
1856 - 360 CONTINUE
1857 - 370 CONTINUE

```

```

1858 - 400 CONTINUE
1859 - C
1860 - C APPLY UDL AT APPROPRIATE NODES. APPLY CONCENTRATED LOAD AT
1861 - C THE EDGE NODE AT EACH FLOOR LEVEL (TOP OF BEAM)
1862 - C
1863 -     WIDTH=W1/DFLOAT(NEH1)
1864 -     NH=NEH1+1
1865 -     INC=NH*(NEV+1)
1866 -     NODE=1+NH
1867 -     DO 430 I=1,NFL
1868 -         BDL=DEN*GAP*BH(1)*PT(1)/2.000
1869 -         NODE=NODE+INC
1870 -         DO 430 J=1,NH
1871 -             IF(J.EQ.1) WRITE(4,90,REC=NODE) PLOAD(1)
1872 -             IF(J.EQ.NH) THEN
1873 -                 READ(3,90,REC=NODE) WGHT
1874 -                 WGHT=WGHT+BDL
1875 -                 WRITE(3,90,REC=NODE) WGHT
1876 -             ELSE
1877 -                 END IF
1878 -             IF((J.EQ.1).OR.(J.EQ.NH)) THEN
1879 -                 F1=0.500
1880 -             ELSE
1881 -                 F1=1.000
1882 -             END IF
1883 -             READ(3,90,REC=NODE) WGHT
1884 -             WGHT=WGHT+UDL(1)*WIDTH*F1
1885 -             WRITE(3,90,REC=NODE) WGHT
1886 -             NODE=NODE+1
1887 - 430 CONTINUE
1888 - 450 CONTINUE
1889 -     WIDTH=W2/DFLOAT(NEH2)
1890 -     NH=NEH2+1
1891 -     INC=NH*(NEV+1)
1892 -     NODE=NODE+NH
1893 -     DO 500 I=1,NFL
1894 -         BDL=DEN*GAP*BH(1)*PT(1)/2.000
1895 -         NODE=NODE+INC
1896 -         DO 480 J=1,NH
1897 -             IF(J.EQ.1) THEN
1898 -                 READ(3,90,REC=NODE) WGHT
1899 -                 WGHT=WGHT+BDL
1900 -                 WRITE(3,90,REC=NODE) WGHT
1901 -             ELSE
1902 -                 END IF
1903 -             IF((J.EQ.1).OR.(J.EQ.NH)) THEN
1904 -                 F1=0.500
1905 -             ELSE
1906 -                 F1=1.000
1907 -             END IF
1908 -             READ(3,90,REC=NODE) WGHT
1909 -             WGHT=WGHT+UDL(1)*WIDTH*F1
1910 -             WRITE(3,90,REC=NODE) WGHT
1911 -             NODE=NODE+1
1912 - 480 CONTINUE
1913 - 500 CONTINUE
1914 - C
1915 - C LATERAL DISTRIBUTED LOAD
1916 - C
1917 -     IF((BLOAD.EQ.0.0,DO).AND.(TLOAD.EQ.0.0,DO)) GOTO 610
1918 -     BL=BLOAD
1919 -     INC=NEH1+1
1920 -     NV=NEV+1
1921 -     NODE=INC+1
1922 -     HT=0.00
1923 -     DO 510 I=1,NFL
1924 -         HT=HT+NH(1)*BH(1)
1925 - 510 CONTINUE
1926 -     FRAC=(TLOAD-BLOAD)/HT

```



```

1927 - DO 600 I=1,NFL
1928 - H1=WH(I)/DFLOAT(NEV)
1929 - H2=BH(I)/2.000
1930 - NODE=NODE+INC
1931 - DO 550 J=1,NV
1932 - IF((J.EQ.1).OR.(J.EQ.NV)) THEN
1933 - F1=0.500
1934 - ELSE
1935 - F1=1.000
1936 - END IF
1937 - TL=FRAC*H1*F1+BL
1938 - READ(4,90,REC=NODE) FORCE
1939 - FORCE=FORCE+H1*F1*(BL+TL)/2.000
1940 - WRITE(4,90,REC=NODE) FORCE
1941 - NODE=NODE+INC
1942 - BL=TL
1943 - 550 CONTINUE
1944 - NODE=NODE+INC
1945 - DO 570 J=1,3
1946 - IF(J.EQ.2) THEN
1947 - F1=1.000
1948 - ELSE
1949 - F1=0.500
1950 - END IF
1951 - TL=FRAC*H2*F1+BL
1952 - READ(4,90,REC=NODE) FORCE
1953 - FORCE=FORCE+H2*F1*(BL+TL)/2.000
1954 - WRITE(4,90,REC=NODE) FORCE
1955 - NODE=NODE+INC
1956 - BL=TL
1957 - 570 CONTINUE
1958 - 600 CONTINUE
1959 - C
1960 - C WRITE LOADS FROM FILES 3 AND 4 TO STARDYNE INPUT FILE 1
1961 - C
1962 - 610 CONTINUE
1963 - DO 630 I=1,NN2-1
1964 - READ(3,90,REC=1) VLOAD
1965 - READ(4,90,REC=1) HLOAD
1966 - WRITE(1,620) I,HLOAD,-VLOAD
1967 - 620 FORMAT('CONC',18,2F10.3)
1968 - 630 CONTINUE
1969 - DO 640 I=NN2,NEND
1970 - READ(3,90,REC=1) VLOAD
1971 - WRITE(1,620) I,0.00,-VLOAD
1972 - 640 CONTINUE
1973 - WRITE(1,200)
1974 - C
1975 - C WRITE END CARDS AND OUTPUT CARDS
1976 - C
1977 - WRITE(1,650)
1978 - 650 FORMAT('END CASE','ALL DONE')
1979 - WRITE(1,690) 1
1980 - 690 FORMAT('OUTPUT',4X,15,5X,3'2',2X,3'2',8X,2'2')
1981 - WRITE(1,700)
1982 - 700 FORMAT('ENDMODEL')
1983 - C
1984 - C WRITE PARAMETERS WHICH WILL BE PASSED TO POSTPROCESSOR ON
1985 - C FILE 2, AND CLOSE THE FILES AND RETURN.
1986 - C
1987 - WRITE(2,720) NFL,IEL,NN2,NEH1,NEH2,NEV
1988 - 720 FORMAT('/RAW DATA:',6(1X,16))
1989 - WRITE(2,730) W1,GAP,W2,BLOAD,TLOAD
1990 - 730 FORMAT(5(1X,F9.3))
1991 - DO 750 I=1,NFL
1992 - WRITE(2,740) I,PT(I),WH(I),BH(I),UDL(I),PLOAD(I)
1993 - 740 FORMAT(14,5(1X,F9.3))
1994 - 750 CONTINUE
1995 - WRITE(2,200)

```

```
1996 - CLOSE(UNIT=1,STATUS='KEEP')
1997 - CLOSE(UNIT=2,STATUS='KEEP')
1998 - CLOSE(UNIT=3,STATUS='DELETE')
1999 - CLOSE(UNIT=4,STATUS='DELETE')
2000 - WRITE(108,770) SFILE,CFILE
2001 - 770 FORMAT(//
2002 - +' Geometry and load data for STARDYNE written on file ',A36/
2003 - +' Postprocessor commands written on file ',A36//
2004 - +' ++++ NORMAL PROGRAM TERMINATION ++++'//)
2005 - RETURN
2006 - END
```

Appendix B

Program listing of postprocessor

```

1 - C **** POST - WALL **** P. KALAB; VERSION 2.0; DECEMBER 2, 1983. ****
2 - C POSTPROCESSOR FOR COUPLED SHEAR WALLS WHICH WERE ANALYZED USING THE
3 - C STARDYNE FINITE ELEMENT PROGRAM. CONTROL PARAMETERS FOR POSTPROCESSOR
4 - C WERE CREATED USING "WALL" PREPROCESSOR AND WRITTEN ON FILE #2 WHICH
5 - C MUST BE CONNECTED TO THIS PROGRAM USING THE OPEN STATEMENT.
6 - C
7 - C * VARIABLES USED - ARRAYS *
8 - C
9 - C WT(50) = wall thickness for each storey (m)
10 - C WH(50) = wall height from floor base to bottom of beam (m)
11 - C BH(50) = beam height (m) (WH+BH=storey height)
12 - C UDL(50) = vertical UDL acting on each floor (kN/m)
13 - C PL(50) = concentrated lateral load acting on each floor (kN)
14 - C SL(50) = lateral LDL acting on base and top of each floor (kN/m)
15 - C AX(50) = axial force in beam (kN)
16 - C VA(50) = shear force in beam at JA end (kN)
17 - C VM(50) = shear force in beam at midspan (kN)
18 - C VB(50) = shear force in beam at JB end (kN)
19 - C BMA(50) = bending moment in beam at JA end (kN*m)
20 - C BMM(50) = bending moment in beam at midspan (kN*m)
21 - C BMB(50) = bending moment in beam at JB end (kN*m)
22 - C ISX(50) = element having max. SX stress in storey 1
23 - C ISY(50) = element having max. SY stress in storey 1
24 - C ISXY(50) = element having max. SXY stress in storey 1
25 - C
26 - C * VARIABLES USED - CHARACTERS *
27 - C
28 - C LINE#132 = reads lines from STARDYNE output
29 - C BUF#72 = temporary buffer and variable passed to CN72
30 - C CFILE#36 = file name where preprocessor data is stored
31 - C SFILE#35 = STARDYNE output file name
32 - C RESP#1 = variable for reading "Y","N","S" responses
33 - C
34 - C * VARIABLES USED - INTEGERS *
35 - C
36 - C I,J,K,L = general DO-LOOP counters
37 - C MAXF = maximum number of floors (50 with existing arrays)
38 - C NFL = actual number of floors
39 - C NN2 = node number where Wall 2 starts
40 - C IEL2 = element number where Wall 1 ends
41 - C NEV = number of horizontal element rows per storey
42 - C NEH1 = number of vertical element rows in Wall 1
43 - C NEH2 = number of vertical element rows in Wall 2
44 - C IER = general flag for error conditions
45 - C NODE = node being processed
46 - C IEL = element being processed
47 - C
48 - C * VARIABLES USED - DOUBLE PRECISION *
49 - C
50 - C W1 = width of Wall 1 (m)
51 - C GAP = distance between walls (m)
52 - C W2 = width of Wall 2 (m)
53 - C BLOAD = LDL magnitude at base of Wall 1 (kN/m)
54 - C TLOAD = LDL magnitude at top of Wall 1 (kN/m)
55 - C VINC = vertical height of wall element (m)
56 - C BINC = vertical height of beam element (m)
57 - C HINC = horizontal length of element (m)
58 - C X,Y = element centroid coordinates
59 - C SX,SY = element stresses in X and Y directions, respectively
60 - C SXY = element shear stress (kN/sq.m later converted to MPa)
61 - C SXM = maximum SX stress in storey
62 - C SYM = maximum SY stress in storey
63 - C SXYM = maximum SXY stress in storey

```

```

64 - C
65 - C Any other variables not mentioned are temporary counters, flags or
66 - C storage locations and their use is evident from the program.
67 - C
68 - C * I/O DEVICE NUMBERS *
69 - C
70 - C 108 = default write to terminal
71 - C 105 = default read from terminal
72 - C 1 = STARDYNE output file (sequential, kept)
73 - C 2 = preprocessor command file and hardcopy output file (seq., kept)
74 - C 3 = scratch files for centroids and stresses (direct, deleted)
75 - C
76 - C * SUBROUTINES USED *
77 - C
78 - C OCHECK = checks for errors in opening files
79 - C CN72 = checks for correct numerical input
80 - C Q = writes prompt, reads answer, checks for input errors
81 - C WL = used by Q to calculate string length of prompt
82 - C PDATA = prints processed data in neat form
83 - C TPAR = tests DO-LOOP parameters so they are in proper range
84 - C
85 - C * COMMON STATEMENTS *
86 - C
87 - C MAIN1, MAIN2, MAIN3 shared with PDATA
88 - C
89 - C * RESTRICTIONS *
90 - C
91 - C This program is restricted to analyzing only coupled shear walls for
92 - C buildings which are at most 50 stories high. Taller structures may be
93 - C analyzed if array dimensions and MAXF values are changed, and some
94 - C FORMAT statements may have to be altered as well.
95 - C
96 - C *****
97 - C
98 - C **** * D I S C L A I M E R * ****
99 - C
100 - C ALTHOUGH THIS PROGRAM INCLUDES MANY TESTS TO CATCH INCORRECT DATA, AND
101 - C ALTHOUGH THE PROGRAM HAS BEEN RIGOROUSLY TESTED, IT HAS BEEN INTENDED
102 - C ONLY FOR ILLUSTRATION PURPOSES, AND NEITHER THE AUTHOR, THE DEPARTMENT
103 - C OF CIVIL ENGINEERING, NOR CARLETON UNIVERSITY ASSUME ANY RESPONSIBILI-
104 - C TY FOR ITS USE BY ANY INDIVIDUAL OR ORGANIZATION.
105 - C
106 - C *****
107 - C
108 - C
109 - C IMPLICIT DOUBLE PRECISION(A-H,O-Z)
110 - C IMPLICIT INTEGER(I-N)
111 - C CHARACTER LINE*132,BUF*72,SFILE*36,CFILE*36,RESP*1,WD*132
112 - C DIMENSION WT(50),WH(50),BH(50),UDL(50),PL(50),SL(51)
113 - C DIMENSION AX(50),VA(50),VB(50),BMA(50),BMB(50),DISP(50)
114 - C DIMENSION ISX(50),ISY(50),ISXY(50),VM(50),BMM(50)
115 - C COMMON /MAIN1/ NFL,NEH1,NEH2,NEV
116 - C COMMON /MAIN2/ WT,WH,BH,UDL,PL,SL,DISP
117 - C COMMON /MAIN3/ AX,VA,VB,BMA,BMM,BMB,ISX,ISY,ISXY
118 - C COMMON /MAIN4/ W1,GAP,W2
119 - C MAXF=50
120 - C 1 FORMAT(A1)
121 - C 6 FORMAT(6G)
122 - C 36 FORMAT(A36)
123 - C 72 FORMAT(A72)
124 - C 132 FORMAT(A132)
125 - C WRITE(108,200)
126 - C 200 FORMAT(//
127 - C + ' STARDYNE POSTPROCESSOR FOR COUPLED SHEAR WALLS - METRIC',
128 - C + ' VERSION'//1X,63'//)
129 - C
130 - C OPEN INPUT FILES AND READ CONTROL PARAMETERS
131 - C
132 - C 250 CONTINUE

```

```

133 - PRINT *, 'Enter name of file where STARDYNE output is stored:'
134 - READ(105,36) SFILE
135 - IF(SFILE(1:4).EQ.'STOP') PRINT *, '== ABORTED =='; STOP
136 - CALL OCHECK(1,SFILE,'OLD','SEQUENTIAL',IER)
137 - IF(IER.EQ.1) STOP
138 - PRINT *
139 - + 'Enter name of file where postprocessor commands are stored:'
140 - READ(105,36) CFILE
141 - IF(CFILE(1:4).EQ.'STOP') PRINT *, '== ABORTED =='; STOP
142 - CALL OCHECK(2,CFILE,'OLD','SEQUENTIAL',IER)
143 - IF(IER.EQ.1) STOP
144 - WRITE(108,270)
145 - 270 FORMAT(/// 'Reading data. Please wait ...')
146 - READ(2,72) BUF
147 - LOC=270
148 - WD(1:72)=BUF
149 - REPEAT 300, WHILE (BUF(1:8).NE.'RAW DATA')
150 - READ(2,72,END=9000) BUF
151 - 300 CONTINUE
152 - LOC=300
153 - WD(1:72)=BUF
154 - READ(2,72,END=9000) BUF
155 - CALL CN72(BUF,IER)
156 - IF(IER.EQ.6) THEN
157 - READ(BUF,6) NFL, IELM, NN2, NEH1, NEH2, NEV
158 - ELSE
159 - WRITE(108,330) BUF
160 - 330 FORMAT(' ERROR - Unexpected data encountered: '/IX,A72/
161 - + ' Incorrect file was read or data has been tampered with. '//
162 - + ' == PROCESSING ABORTED ==')
163 - GOTO 9100
164 - END IF
165 - LOC=330
166 - WD(1:72)=BUF
167 - READ(2,72,END=9000) BUF
168 - CALL CN72(BUF,IER)
169 - IF(IER.EQ.5) THEN
170 - READ(BUF,6) W1, GAP, W2, BLOAD, TLOAD
171 - ELSE
172 - WRITE(108,330) BUF
173 - GOTO 9100
174 - END IF
175 - IF(NFL.LT.1) NFL=1
176 - IF(NFL.GT.MAXF) NFL=MAXF
177 - LOC=330
178 - WD(1:72)=BUF
179 - DO 350 I=1,NFL
180 - READ(2,72,END=9000) BUF
181 - CALL CN72(BUF,IER)
182 - IF(IER.EQ.6) THEN
183 - READ(BUF,6) J
184 - IF(J.EQ.1) THEN
185 - READ(BUF,6) J, WT(J), WH(J), BH(J), UDL(J), PL(J)
186 - ELSE
187 - WRITE(108,330) BUF
188 - GOTO 9100
189 - END IF
190 - ELSE
191 - WRITE(108,330) BUF
192 - GOTO 9100
193 - END IF
194 - 350 CONTINUE
195 - LOC=350
196 - WD(1:72)=BUF
197 - READ(2,72,END=9000) BUF
198 - WRITE(2,400)
199 - 400 FORMAT(//33X, I4=' '/33X, 'PROCESSED DATA'/33X, I4=' ')
200 - C
201 - C TRANSFER DATA FROM STARDYNE OUTPUT TO ARRAYS AND SCRATCH FILE

```

```

202 - C
203 - C EXTRACT DISPLACEMENT DATA AND STORE AS STOREY DRIFT IN ARRAY DISP
204 - C
205 -     READ(1,132,END=9000) LINE
206 -     WD=LINE
207 -     LOC=400
208 -     REPEAT 450, WHILE (LINE(57:77).NE.'D E F L E C T I O N S')
209 -         READ(1,132,END=9000) LINE
210 - 450 CONTINUE
211 -     LOC=450
212 -     WD=LINE
213 -     DO 460 I=1,4
214 -         READ(1,132,END=9000) LINE
215 - 460 CONTINUE
216 -     INC=(NEH2+1)*(NEV+2)
217 -     NS=NN2+NEH2+INC
218 -     NODE=1
219 -     DO 500 I=1,NFL
220 -         LOC=460
221 -         WD(1:72)=BUF
222 -         REPEAT 480, WHILE (NODE.NE.NS)
223 -             READ(1,72,END=9000) BUF
224 -             CALL CN72(BUF,IER)
225 -             IF (ABS(IER).GT.2) THEN
226 -                 READ(BUF,6) NODE,X
227 -             ELSE
228 -                 WRITE(108,330) BUF
229 -                 GOTO 9100
230 -             END IF
231 - 480 CONTINUE
232 -         DISP(I)=X*1.D3
233 -         NS=NS+INC
234 - 500 CONTINUE
235 - C
236 - C EXTRACT FORCES ACTING ON BEAM
237 - C
238 -     LOC=500
239 -     WD=LINE
240 -     REPEAT 550, WHILE (LINE(38:44).NE.'B E A M')
241 -         READ(1,132,END=9000) LINE
242 - 550 CONTINUE
243 -     LOC=550
244 -     WD=LINE
245 -     DO 560 I=1,3
246 -         READ(1,132,END=9000) LINE
247 - 560 CONTINUE
248 -     BUF=' '
249 -     LOC=560
250 -     WD=LINE
251 -     DO 600 I=1,NFL
252 -         READ(1,132,END=9000) LINE
253 -         BUF(1:4)=LINE(2:5)
254 -         BUF(5:34)=LINE(23:52)
255 -         BUF(35:49)=LINE(93:107)
256 -         CALL CN72(BUF,IER)
257 -         IF (IER.GT.0) THEN
258 -             READ(BUF,6) J
259 -             IF (J.EQ.1) THEN
260 -                 READ(BUF,6) J,AX(J),VA(J),BMA(J)
261 -             ELSE
262 -                 WRITE(108,330) BUF
263 -                 GOTO 9100
264 -             END IF
265 -         ELSE
266 -             WRITE(108,330) BUF
267 -         END IF
268 -     LOC=562
269 -     WD=LINE
270 -     READ(1,132,END=9000) LINE

```

```

271 -      BUF(1:4)=LINE(2:5)
272 -      BUF(5:20)=LINE(37:52)
273 -      BUF(21:35)=LINE(93:107)
274 -      CALL CN72(BUF,IER)
275 -      IF(IER.GT.0) THEN
276 -        READ(BUF,6) J
277 -        IF(J.EQ.1) THEN
278 -          READ(BUF,6) J,VM(J),BMM(J)
279 -        ELSE
280 -          WRITE(108,330) BUF
281 -          GOTO 9100
282 -        END IF
283 -      ELSE
284 -        WRITE(108,330) BUF
285 -      END IF
286 -      READ(1,132,END=9000) LINE
287 -      READ(1,132,END=9000) LINE
288 -      BUF(1:4)=LINE(2:5)
289 -      BUF(5:20)=LINE(37:52)
290 -      BUF(21:35)=LINE(93:107)
291 -      CALL CN72(BUF,IER)
292 -      IF(IER.GT.0) THEN
293 -        READ(BUF,6) J
294 -        IF(J.EQ.1) THEN
295 -          READ(BUF,6) J,VB(J),BMB(J)
296 -        ELSE
297 -          WRITE(108,330) BUF
298 -          GOTO 9100
299 -        END IF
300 -      ELSE
301 -        WRITE(108,330) BUF
302 -      END IF
303 -      600 CONTINUE
304 - C
305 - C EXTRACT ELEMENT STRESSES, CONVERT FROM KN/SQ MM TO MPa, AND WRITE
306 - C TO FILE 3 ALONG WITH THE COORDINATES.
307 - C
308 -      OPEN(UNIT=3,NAME='*F3',STATUS='SCRATCH',ACCESS='DIRECT',
309 -      + FORM='FORMATTED')
310 -      DO 640 I=1,IELM
311 -        WRITE(3,630,REC=1) 0.0,0.0,9.999E+99,9.999E+99,9.999E+99
312 - 630      FORMAT(2(F9.4,1X),3E13.4)
313 - 640 CONTINUE
314 -      REPEAT 650, WHILE (LINE(56:70).NE.'S T R E S S E S')
315 -        READ(1,132,END=9000) LINE
316 - 650 CONTINUE
317 -      LOC=650
318 -      WD=LINE
319 -      READ(1,132,END=9000) LINE
320 -      READ(1,132,END=9000) LINE
321 -      IEL=1
322 -      HINC=W1/DFLOAT(NEH1)
323 -      XS=HINC/2.00
324 -      Y=0.00
325 -      DO 800 I=1,NFL
326 -        VINC=WH(1)/DFLOAT(NEV)
327 -        BINC=BH(1)/2.00
328 -        Y=Y-VINC/2.00
329 -        DO 760 J=1,NEV
330 -          X=XS
331 -          Y=Y+VINC
332 -          DO 740 K=1,NEH1
333 -            WRITE(3,720,REC=IEL) X,Y
334 - 720          FORMAT(2(F9.4,1X))
335 -            IEL=IEL+1
336 -            X=X+HINC
337 -          740 CONTINUE
338 -        760 CONTINUE
339 -        Y=Y+(VINC-BINC)/2.00

```

```

340 -      DO 780 J=1,2
341 -          X=XS
342 -          Y=Y+BINC
343 -          DO 770 K=1,NEH1
344 -              WRITE(3,720,REC=IEL) X,Y
345 -              IEL=IEL+1
346 -              X=X+HINC
347 - 770      CONTINUE
348 - 780      CONTINUE
349 -          Y=Y+BINC/2,DO
350 - 800      CONTINUE
351 -          HINC=H2/DFLOAT(NEH2)
352 -          XS=HINC/2,DO
353 -          Y=0,DO
354 -          DO 900 I=1,NFL
355 -              VINC=WH(1)/DFLOAT(NEV)
356 -              BINC=BH(1)/2,DO
357 -              Y=Y-VINC/2,DO
358 -              DO 860 J=1,NEV
359 -                  X=XS
360 -                  Y=Y+VINC
361 -                  DO 840 K=1,NEH2
362 -                      WRITE(3,720,REC=IEL) X,Y
363 -                      IEL=IEL+1
364 -                      X=X+HINC
365 - 840          CONTINUE
366 - 860          CONTINUE
367 -          Y=Y+(VINC-BINC)/2,DO
368 -          DO 880 J=1,2
369 -              X=XS
370 -              Y=Y+BINC
371 -              DO 870 K=1,NEH2
372 -                  WRITE(3,720,REC=IEL) X,Y
373 -                  IEL=IEL+1
374 -                  X=X+HINC
375 - 870          CONTINUE
376 - 880          CONTINUE
377 -          Y=Y+BINC/2,DO
378 - 900          CONTINUE
379 - 910          CONTINUE
380 -          BUF=' '
381 -          LOC=910
382 -          WD=LINE
383 -          REPEAT 920, WHILE (LINE(41:46).NE.'-ZFACE')
384 -              READ(1,132,END=9000) LINE
385 -              IF(LINE(1:1).EQ.'1') GOTO 992
386 - 920          CONTINUE
387 -          LOC=920
388 -          WD=LINE
389 -          BUF(1:5)=LINE(9:13)
390 -          BUF(8:20)=LINE(52:64)
391 -          READ(1,132,END=9000) LINE
392 -          BUF(21:33)=LINE(52:64)
393 -          READ(1,132,END=9000) LINE
394 -          BUF(34:46)=LINE(52:64)
395 -          CALL CN72(BUF,IER)
396 -          IF((IER.GE.4).OR.(IER.LT.-4)) THEN
397 -              READ(BUF,6) IC,SX,SY,SXY
398 -          ELSE IF((IER.GE.3).OR.(IER.LT.-3)) THEN
399 -              READ(BUF,6) IC,SX,SY
400 -              SXY=9,9990+99
401 -              PRINT *, 'ERROR in reading SXY of element ',IC
402 -          ELSE IF((IER.GE.2).OR.(IER.LT.-2)) THEN
403 -              READ(BUF,6) IC,SX
404 -              SXY=9,9990+99
405 -              SY=9,9990+99
406 -              PRINT *, 'ERROR in reading SXY and SY of element ',IC
407 -          ELSE IF((IER.GE.1).OR.(IER.LT.-1)) THEN
408 -              READ(BUF,6) IC

```



```

409 -      SXY=9.999D+99
410 -      SY=9.999D+99
411 -      SX=9.999D+99
412 -      PRINT *, 'ERROR in reading SXY, SY, and SX of element ', IC
413 -      ELSE
414 -      PRINT *, 'ERROR in reading all data for element ', IEL
415 -      GOTO 990
416 -      END IF
417 -      IF (IC.LE.0) THEN
418 -      WRITE(108,922) IC,WD(1:72)
419 -      922 FORMAT(
420 -      + ' ERROR - Element no. read is less than 1:',15/
421 -      + ' Last correct element read was:',1/A72)
422 -      GOTO 990
423 -      ELSE IF (IC.GT.IELM) THEN
424 -      WRITE(108,924) IC,WD(1:72)
425 -      924 FORMAT(
426 -      + ' ERROR - Element no. read is larger than expected:',15/
427 -      + ' Last correct element read was:',1/A72)
428 -      GOTO 990
429 -      ELSE
430 -      END IF
431 -      LOC=924
432 -      WD=LINE
433 -      READ(3,6,REC=IC) X,Y
434 -      WRITE(3,630,REC=IC) X,Y,SX/1.D3,SY/1.D3,SXY/1.D3
435 -      990 CONTINUE
436 -      GOTO 910
437 -      992 CONTINUE
438 -      C
439 -      C CALCULATE MAXIMUM STRESSES IN EACH FLOOR AND STORE THE ELEMENT
440 -      C WHERE EACH STRESS OCCURS IN INDEX ARRAY.
441 -      C
442 -      IEL2=NFL*(NEV+2)*NEH1
443 -      DO 1005 I=1,NFL
444 -      .SXM=0.D0
445 -      SYM=0.D0
446 -      SXYM=0.D0
447 -      DO 1003 J=1,NEV+2
448 -      DO 1001 K=1,NEH1
449 -      L=(I-1)*(NEV+2)*NEH1+(J-1)*NEH1+K
450 -      READ(3,1000,REC=L) X,Y,SX,SY,SXY
451 -      1000 FORMAT(5G)
452 -      IF ((DABS(SX).GT.ABS(SXM)).AND.(SX.NE.9.999D+96))
453 -      + ISX(I)=L ; SXM=SX
454 -      IF ((DABS(SY).GT.ABS(SYM)).AND.(SY.NE.9.999D+96))
455 -      + ISY(I)=L ; SYM=SY
456 -      IF ((DABS(SXY).GT.ABS(SXYM)).AND.(SXY.NE.9.999D+96))
457 -      + ISXY(I)=L ; SXYM=SXY
458 -      1001 CONTINUE
459 -      DO 1002 K=1,NEH2
460 -      L=(I-1)*(NEV+2)*NEH2+(J-1)*NEH2+K+IEL2
461 -      READ(3,1000,REC=L) X,Y,SX,SY,SXY
462 -      IF ((DABS(SX).GT.ABS(SXM)).AND.(SX.NE.9.999D+96))
463 -      + ISX(I)=L ; SXM=SX
464 -      IF ((DABS(SY).GT.ABS(SYM)).AND.(SY.NE.9.999D+96))
465 -      + ISY(I)=L ; SYM=SY
466 -      IF ((DABS(SXY).GT.ABS(SXYM)).AND.(SXY.NE.9.999D+96))
467 -      + ISXY(I)=L ; SXYM=SXY
468 -      1002 CONTINUE
469 -      1003 CONTINUE
470 -      1005 CONTINUE
471 -      C
472 -      C CALCULATE TOP AND BOTTOM VALUES OF LDL ACTING ON FLOOR
473 -      C
474 -      HT=0.D0
475 -      DO 1010 I=1,NFL
476 -      HT=HT+WH(I)+BH(I)
477 -      1010 CONTINUE

```

```

478 -      FRAC=(TLOAD-BLOAD)/HT
479 -      SL(1)=BLOAD
480 -      DO 1020 I=1,NFL
481 -          SL(I+1)=SL(I)+FRAC*(WH(I)+BH(I))
482 - 1020 CONTINUE
483 -      CALL PDATA
484 -      GOTO 9100
485 - 9000 CONTINUE
486 -      WRITE(108,9010) LOC,WD
487 - 9010 FORMAT(/' ERROR - End of file reached and data was not found. '/
488 -      +' Incorrect file was opened or data is incomplete. '/
489 -      +' Preceding statement no. in source code =',15/
490 -      +' Last valid statement read: '/A132)
491 - 9100 CONTINUE
492 -      CLOSE(UNIT=1,STATUS='KEEP')
493 -      CLOSE(UNIT=2,STATUS='KEEP')
494 -      CLOSE(UNIT=3,STATUS='DELETE')
495 -      WRITE(108,9105) CFILE
496 - 9105 FORMAT(/' Processed data have been written on file ',A36)
497 - 9110 CONTINUE
498 -      WRITE(108,9120)
499 - 9120 FORMAT(/' Do you want to process different data (Y/N)?')
500 -      READ(105,1) RESP
501 -      IF(RESP.EQ.'Y') THEN
502 -          GOTO 250
503 -      ELSE IF(RESP.EQ.'N') THEN
504 -          WRITE(108,9130)
505 - 9130 FORMAT(/' ++++ NORMAL TERMINATION ++++')
506 -          STOP
507 -      ELSE
508 -          PRINT *, 'Please answer "Y" or "N":'
509 -          GOTO 9110
510 -      END IF
511 -      END
512 - C
513 - C *****
514 - C **** PDATA **** P. KALAB; VERSION 1.0; DECEMBER 3, 1983. ****
515 - C SUBROUTINE WHICH PRINTS THE APPROPRIATE DATA REQUESTED BY THE USER.
516 - C * SYNTAX * PDATA
517 - C * CALL BY * POST
518 - C * CALLS * CN72,Q,TPAR
519 - C * COMMON * MAIN1(POST), MAIN2(POST), MAIN3(POST)
520 - C
521 -      SUBROUTINE PDATA
522 -      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
523 -      IMPLICIT INTEGER(I-N)
524 -      CHARACTER RESP*1, ANS*72
525 -      DIMENSION WT(50),WH(50),BH(50),UDL(50),PL(50),DISP(50)
526 -      DIMENSION AX(50),VA(50),VB(50),BMA(50),BMB(50),SL(51)
527 -      DIMENSION ISX(50),ISY(50),ISXY(50),VM(50),BMM(50)
528 -      COMMON /MAIN1/ NFL,NEH1,NEH2,NEV
529 -      COMMON /MAIN2/ WT,WH,BH,UDL,PL,SL,DISP
530 -      COMMON /MAIN3/ AX,VA,VM,VB,BMA,BMM,BMB,ISX,ISY,ISXY
531 -      COMMON /MAIN4/ W1,GAP,W2
532 -      NEV=NEV+2
533 -      1 FORMAT(A1)
534 -      3 FORMAT(3G)
535 - 100 CONTINUE
536 -      WRITE(2,110)
537 - 110 FORMAT(1X,79' ')
538 - 120 FORMAT(/1X,79' ')
539 - 130 CONTINUE
540 -      WRITE(108,140)
541 - 140 FORMAT(/' Do you want all data printed (Y/N)?')
542 -      READ(105,1) RESP
543 -      IF(RESP.EQ.'Y') THEN
544 -          IFS=1
545 -          IFE=NFL
546 -          IFI=1

```

```

547 -      IVS=1
548 -      IVE=NEV
549 -      IVI=1
550 -      IHIS=1
551 -      IHIE=NEH1
552 -      IH1I=1
553 -      IH2S=1
554 -      IH2E=NEH2
555 -      IH2I=1
556 -      ELSE IF(RESPEQ,'S') THEN
557 -        PRINT *, '== PROCESSING ABORTED =='
558 -        STOP
559 -      ELSE IF(RESPEQ,'N') THEN
560 -        WRITE(108,160)
561 -        160 FORMAT(/
562 -        + 'After the following prompts, enter the starting value, end'/
563 -        + 'value, and increment for the information you want'
564 -        + 'processed.')
565 -        WRITE(108,170) NFI
566 -        170 FORMAT(/ 'Structure has',13,' storeys.')
567 -        CALL Q(
568 -        + 'Enter start, end, and increment for storeys processed:',
569 -        + 3,ANS)
570 -        READ(ANS,3) IFS,IFE,IFI
571 -        CALL TPAR(IFS,IFE,IFI,NFI)
572 -        WRITE(108,190) NEV
573 -        190 FORMAT(/ 'Each storey has',15,' elements in vertical',
574 -        + 'direction.')
575 -        CALL Q(
576 -        + 'Enter start, end, and increment for elements processed:',
577 -        + 3,ANS)
578 -        READ(ANS,3) IVS,IVE,IVI
579 -        CALL TPAR(IVS,IVE,IVI,NEV)
580 -        WRITE(108,200) NEH1
581 -        200 FORMAT(/ 'Wall 1 has',15,' elements in horizontal',
582 -        + 'direction.')
583 -        CALL Q(
584 -        + 'Enter start, end, and increment for elements processed:',
585 -        + 3,ANS)
586 -        READ(ANS,3) IHIS,IHIE,IH1I
587 -        CALL TPAR(IHIS,IHIE,IH1I,NEH1)
588 -        WRITE(108,210) NEH2
589 -        210 FORMAT(/ 'Wall 2 has',15,' elements in horizontal',
590 -        + 'direction.')
591 -        CALL Q(
592 -        + 'Enter start, end, and increment for elements processed:',
593 -        + 3,ANS)
594 -        READ(ANS,3) IH2S,IH2E,IH2I
595 -        CALL TPAR(IH2S,IH2E,IH2I,NEH2)
596 -      ELSE
597 -        PRINT *, 'Please answer "Y", "N", or "S" to abort.'
598 -        GOTO 130
599 -      END IF
600 -      IEL2=NFI*NEV*NEH1
601 - C
602 - C PRINT DATA IN REARRANGED FORM
603 - C
604 -      DO 550 I=IFS,IFE,IFI
605 -        WRITE(108,120)
606 -        WRITE(2,120)
607 -        WRITE(108,250) I,DISP(I),W1,W2,GAP,BH(I),WH(I)+BH(I),WT(I)
608 -        WRITE(2,250) I,DISP(I),W1,W2,GAP,BH(I),WH(I)+BH(I),WT(I)
609 -        250 FORMAT(/ 'X,25'// 'X,5'// ' STOREY NO:',13,'X,5'//
610 -        + 5X,'Storey drift =',F9.3,' mm'// 'X,25'//
611 -        + ' GEOMETRY:')
612 -        + ' Wall 1 width =',F7.3,' m   Wall 2 width =',F7.3,' m'//
613 -        + ' Beam length =',F7.3,' m   Beam height =',F7.3,' m'//
614 -        + ' Storey height =',F7.3,' m   Thickness =',F7.3,' m'//
615 -        WRITE(108,255) SL(I+1),SL(I),PL(I),UDL(I)

```

```

616 -      WRITE(2,255) SL(1+1),SL(1),PL(1),UOL(1)
617 - 255   FORMAT(' LOADS: '/)
618 -      + ' Lateral top LDL =' ,F9.3, ' kN/m   Bot. LDL =' ,F9.3, ' kN/m' /
619 -      + ' Conc. lat. load =' ,F9.3, ' kN       Vert. UOL =' ,F9.3, ' kN/m' //)
620 -      WRITE(108,260) AX(1),AX(1),-AX(1),VA(1),VM(1),VB(1),BMA(1),
621 -      + BMM(1),BMB(1)
622 -      WRITE(2,260) AX(1),AX(1),-AX(1),VA(1),VM(1),VB(1),BMA(1),
623 -      + BMM(1),BMB(1)
624 - 260   FORMAT(' FORCES IN CONNECTING BEAM: '/)
625 -      + 17X, 'At Wall 1', 7X, 'At midspan', 7X, 'At Wall 2' /
626 -      + ' Axial (kN)', 3X, F11.3, 4X, F12.3, 5X, F12.3 /
627 -      + ' Shear (kN)', 3X, F11.3, 4X, F12.3, 5X, F12.3 /
628 -      + ' Moment (kN*m)', F11.3, 4X, F12.3, 5X, F12.3 //)
629 -      WRITE(108,262)
630 -      WRITE(2,262)
631 - 262   FORMAT(' MAXIMUM STRESS SUMMARY (in MPa): ' //
632 -      + 6X, 'X', 11X, 'Y', 10X, 'XY', 7X, 'ELEMENT' /)
633 -      READ(3,350,REC=1SX(1),ERR=700) X,Y,SX,SY,SXY
634 -      WRITE(108,263) SX,1SX(1)
635 -      WRITE(2,263) SX,1SX(1)
636 - 263   FORMAT(1X,E10.4,26X,16)
637 -      READ(3,350,REC=1SY(1),ERR=700) X,Y,SX,SY,SXY
638 -      WRITE(108,264) SY,1SY(1)
639 -      WRITE(2,264) SY,1SY(1)
640 - 264   FORMAT(11X,E12.4,14X,16)
641 -      READ(3,350,REC=1SXY(1),ERR=700) X,Y,SX,SY,SXY
642 -      WRITE(108,265) SXY,1SXY(1)
643 -      WRITE(2,265) SXY,1SXY(1)
644 - 265   FORMAT(25X,E10.4,18/)
645 -      WRITE(108,266)
646 -      WRITE(2,266)
647 - 266   FORMAT(1X,79' ')
648 -      WRITE(108,270)
649 -      WRITE(108,280)
650 -      WRITE(2,270)
651 -      WRITE(2,280)
652 - 270   FORMAT('/' WALL 1: '/1X,7'=' /)
653 - 280   FORMAT(
654 -      + ' ELEMENT', 7X, 'LOCATION', 22X, 'STRESSES' /
655 -      + 14X, 'X', 8X, 'Y', 12X, 'X', 11X, 'Y', 10X, 'XY' /
656 -      + 13X, '(m)', 6X, '(m)', 9X, '(MPa)', 7X, '(MPa)', 7X, '(MPa)' /)
657 -      DO 450 J=1VS,1VE,1VI
658 -          DO 420 K=1H1S,1H1E,1H1I
659 -              L=(1-1)*NEV*NEH1+(J-1)*NEH1+K
660 -              READ(3,350,REC=L,ERR=700) X,Y,SX,SY,SXY
661 - 350     FORMAT(5G)
662 -          WRITE(108,400) L,X,Y,SX,SY,SXY
663 -          WRITE(2,400) L,X,Y,SX,SY,SXY
664 - 400     FORMAT(16,3X,2F9.3,2X,3(2X,E10.4))
665 - 420     CONTINUE
666 - 450     CONTINUE
667 -          WRITE(108,460)
668 -          WRITE(2,460)
669 - 460     FORMAT('/' WALL 2: '/1X,7'=' /)
670 -          WRITE(108,280)
671 -          WRITE(2,280)
672 -          DO 520 J=1VS,1VE,1VI
673 -              DO 500 K=1H2S,1H2E,1H2I
674 -                  L=(1-1)*NEV*NEH2+(J-1)*NEH2+K+1EL2
675 -                  READ(3,350,REC=L,ERR=700) X,Y,SX,SY,SXY
676 -                  WRITE(108,400) L,X,Y,SX,SY,SXY
677 -                  WRITE(2,400) L,X,Y,SX,SY,SXY
678 - 500     CONTINUE
679 - 520     CONTINUE
680 - 550     CONTINUE
681 -          WRITE(108,120)
682 -          WRITE(2,120)
683 - 560     CONTINUE
684 -          WRITE(108,600)

```

```

685 - 600 FORMAT(//
686 -      +1 Do you wish to display data at different increments (Y/N)?')
687 -      READ(105,1) RESP
688 -      IF(RESP.EQ.'Y') THEN
689 -          GOTO 100
690 -      ELSE IF(RESP.EQ.'N') THEN
691 -          RETURN
692 -      ELSE IF(RESP.EQ.'S') THEN
693 -          PRINT *, '== ABNORMAL TERMINATION =='
694 -          STOP
695 -      ELSE
696 -          PRINT *, 'Please answer "Y", "N", or "S" to abort.'
697 -          GOTO 560
698 -      END IF
699 - 700 CONTINUE
700 -      PRINT *,
701 -      +1 ERROR in trying to read missing record ',L
702 -      GOTO 560
703 -      END
704 - C
705 - C *****
706 - C **** TPAR **** P. KALAB; VERSION 1.0; DECEMBER 2, 1983. ****
707 - C SUBROUTINE TO TEST INITIALIZATION PARAMETERS FOR DO-LOOPS TO MAKE
708 - C SURE THEY ARE IN PROPER RANGE AND SEQUENCE.
709 - C * SYNTAX * TPAR(START,END,INC,MAX)
710 - C          START = STARTING VALUE OF DO-LOOP COUNTER
711 - C          END   = FINAL VALUE OF DO-LOOP COUNTER
712 - C          INC   = INCREMENT COUNTER BY THIS AMOUNT
713 - C          MAX   = END IS NOT TO EXCEED THIS VALUE
714 - C * CALL BY * PDATA
715 - C
716 - C          SUBROUTINE TPAR(START,END,INC,MAX)
717 - C              INTEGER START,END,INC,MAX
718 - C              IF(START.LE.0) START=1
719 - C              IF((END.LE.0).OR.(END.GT.MAX)) END=MAX
720 - C              IF(INC.LE.0) INC=1
721 - C              IF(START.GT.MAX) START=MAX
722 - C              IF(START.GT.END) END=START
723 - C              RETURN
724 - C          END
725 - C
726 - C *****
727 - C **** Q **** P. KALAB; VERSION 2.0; OCTOBER 28, 1983. ****
728 - C SUBROUTINE WHICH PRINTS A QUESTION PROMPT FOR THE USER AND READS
729 - C THE ANSWER AS A CHARACTER STRING, IT CHECKS WHETHER THE ANSWER HAS
730 - C THE PROPER NUMBER OF NUMERIC ARGUMENTS, AND SETS A FLAG IF IT DOES
731 - C NOT. IF SPECIAL COMMANDS ARE ENTERED SUBROUTINE PERFORMS FOLLOWING:
732 - C STOP = PROGRAM STOPS
733 - C GEOM = IER IS SET TO 1 SO ALL GEOMETRY PARAMETERS MAY BE RE-ENTERED
734 - C LOAD = IER IS SET TO 2 SO ALL LOAD PARAMETERS MAY BE RE-ENTERED
735 - C
736 - C * SYNTAX * Q(QUEST*NARG,ANS*72)
737 - C          QUEST = QUESTION PROMPT PRINTED TO USER
738 - C          NARG  = NO. OF NUMERIC ARGUMENTS EXPECTED
739 - C          ANS   = ANSWER READ AND PASSED BACK AS CHARACTER STRING
740 - C * CALL BY * WALL
741 - C * CALLS * WL, CN72
742 - C
743 - C
744 - C          SUBROUTINE Q(QUEST,NARG,ANS)
745 - C              IMPLICIT REAL(A-H,O-Z)
746 - C              CHARACTER QUEST*72,ANS*72
747 - C              CALL WL(QUEST,NQUEST)
748 - C              ITRY=0
749 - C              10 CONTINUE
750 - C              IF(QUEST(1:4).NE.' ') WRITE(108,30) NQUEST,QUEST
751 - C              30 FORMAT(1X,AN)
752 - C              READ '(A72)',ANS
753 - C              CALL CN72(ANS,IND)

```

```

754 - IF(IND.EQ.0) RETURN
755 - IF((ABS(IND).GE.NARG).AND.(IND.NE.-NARG)) THEN
756 - RETURN
757 - ELSE IF(ANS(1:4).EQ.'STOP') THEN
758 - PRINT *, '== RUN ABORTED =='
759 - STOP
760 - ELSE
761 - ITRY=ITRY+1
762 - IF(ITRY.GE.3) THEN
763 - PRINT *, 'Unable to read answer - returning zero values.'
764 - ANS=' '
765 - RETURN
766 - ELSE
767 - PRINT *, '
768 - +Incorrect numerical value or too few arguments entered:'
769 - PRINT *, ANS
770 - GOTO 10
771 - END IF
772 - END IF
773 - END
774 - C
775 - C *****
776 - C **** WL **** P. KALAB; VERSION 1.0; JANUARY 15, 1983. ****
777 - C SUBROUTINE IS USED TO FIND THE END OF A QUESTION (DESIGNATED BY
778 - C '?', ':', OR ' ') PASSED IN ARGUMENT 'STRING' AND RETURNS
779 - C THE LENGTH IN ARGUMENT 'L'.
780 - C * SYNTAX * WL(STRING*72,L)
781 - C * CALL BY * Q
782 - C
783 - SUBROUTINE WL(STRING,L)
784 - CHARACTER STRING*72
785 - IMPLICIT INTEGER(I-N)
786 - L=1
787 - 10 IF((STRING(L:L).EQ.'?').OR.(STRING(L:L).EQ.':').
788 - + .OR.(L.EQ.72).OR.(STRING(L:L+3).EQ.' ')) RETURN
789 - L=L+1
790 - GOTO 10
791 - END
792 - C
793 - C *****
794 - C **** CN72 **** P. KALAB; VERSION 1.2; JANUARY 25, 1983. ****
795 - C SUBROUTINE RECEIVES AN ALPHANUMERIC STRING IN 'WORD' AND PASSES
796 - C BACK THE NUMBER OF ERROR-FREE NUMERIC FIELDS IN 'ABS(IND)'.
797 - C IF ERRORS ARE ENCOUNTERED, 'IND' BECOMES NEGATIVE.
798 - C * SYNTAX * CN72(CHAR*72,IND)
799 - C CHAR = STRING SCANNED FOR NUMERICAL INPUT
800 - C IND = INTEGER INDICATING ERRORS ENCOUNTERED
801 - C * CALL BY * END
802 - C
803 - SUBROUTINE CN72(WORD,IND)
804 - CHARACTER WORD*72,N1*10
805 - DATA N1/'0123456789'/
806 - IER=0
807 - IEL=72
808 - L=0
809 - IND=-1
810 - 10 CONTINUE
811 - IF(IER.EQ.0) THEN
812 - IND=IND+1
813 - ELSE
814 - IF(IND.EQ.0) IND=-1; RETURN
815 - IND=-1*ABS(IND)
816 - GOTO 600
817 - END IF
818 - ISGN=0
819 - IDOT=0
820 - 20 CONTINUE
821 - L=L+1
822 - IF(WORD(L:L).EQ.' ') THEN

```

```

823 -      IF(L,GE,IEL) GOTO 600
824 -      GOTO 20
825 -      ELSE
826 -      IF(L,GE,IEL) GOTO 600
827 -      END IF
828 -      IF((WORD(L:L),EQ,'+')OR.(WORD(L:L),EQ,'-')) THEN
829 -      ISIGN=1
830 -      30 CONTINUE
831 -      L=L+1
832 -      IF(L,GE,IEL) GOTO 500
833 -      IF(WORD(L:L),EQ,' ') GOTO 30
834 -      IF(WORD(L:L),EQ,'.') THEN
835 -      IDOT=1
836 -      L=L+1
837 -      IF(L,GE,IEL) GOTO 500
838 -      ELSE
839 -      END IF
840 -      DO 40 I=1,10
841 -      IF(WORD(L:L),EQ,N1(I:1)) GOTO 100
842 -      40 CONTINUE
843 -      GOTO 500
844 -      ELSE IF(WORD(L:L),EQ,'') THEN
845 -      GOTO 10
846 -      ELSE IF(WORD(L:L),EQ,'.') THEN
847 -      ISIGN=1
848 -      IDOT=1
849 -      L=L+1
850 -      IF(L,GE,IEL) GOTO 500
851 -      DO 50 I=1,10
852 -      IF(WORD(L:L),EQ,N1(I:1)) GOTO 100
853 -      50 CONTINUE
854 -      GOTO 500
855 -      ELSE
856 -      DO 60 I=1,10
857 -      IF(WORD(L:L),EQ,N1(I:1)) GOTO 100
858 -      60 CONTINUE
859 -      GOTO 500
860 -      END IF
861 -      100 CONTINUE
862 -      L=L+1
863 -      IF(L,GE,IEL) GOTO 600
864 -      DO 110 I=1,10
865 -      IF(WORD(L:L),EQ,N1(I:1)) GOTO 100
866 -      110 CONTINUE
867 -      IF((WORD(L:L),EQ,'')OR.(WORD(L:L),EQ,' ')) GOTO 10
868 -      IF((WORD(L:L),EQ,'.')AND.(IDOT,EQ,0)) THEN
869 -      IDOT=1
870 -      GOTO 100
871 -      ELSE
872 -      END IF
873 -      IF(WORD(L:L),EQ,'E') THEN
874 -      ISIGN=0
875 -      120 CONTINUE
876 -      L=L+1
877 -      IF(WORD(L:L),EQ,' ') GOTO 120
878 -      IF(((WORD(L:L),EQ,'+')OR.(WORD(L:L),EQ,'-'))AND.
879 -      (ISIGN,EQ,0)) THEN
880 -      ISIGN=1
881 -      GOTO 120
882 -      ELSE
883 -      END IF
884 -      DO 130 I=1,10
885 -      IF(WORD(L:L),EQ,N1(I:1)) GOTO 140
886 -      130 CONTINUE
887 -      GOTO 500
888 -      140 CONTINUE
889 -      L=L+1
890 -      IF((WORD(L:L),EQ,'')OR.(WORD(L:L),EQ,' ')) GOTO 10
891 -      DO 150 I=1,10

```

```

892 -         IF(WORD(L:L),EQ,NI(1:1)) GOTO 160
893 - 150     CONTINUE
894 -         GOTO 500
895 - 160     CONTINUE
896 -         L=L+1
897 -         IF(L,GE,IEL) GOTO 600
898 -         IF((WORD(L:L),EQ,' '),OR,(WORD(L:L),EQ,' ')) GOTO 10
899 -         ELSE
900 -         END IF
901 - 500     CONTINUE
902 -         IER=IER+1
903 -         GOTO 10
904 - 600     CONTINUE
905 -         IF(IND,LT,0) IND=IND-1
906 -         RETURN
907 -         END
908 - C
909 - C *****
910 - C **** OCHECK **** P. KALAB; VERSION 1.2; SEPTEMBER 19, 1983. ****
911 - C SUBROUTINE WHICH OPENS FILES AND CHECKS FOR ERROR CONDITIONS. THE
912 - C USER HAS 3 CHANCES TO CORRECT ANY MISTAKES HE MAY HAVE MADE.
913 - C * SYNTAX * OCHECK(I1,CHAR1*36,CHAR2*7,CHAR3*7,12)
914 - C * VARIABLES * I1 = DEVICE NUMBER; CHAR1 = FILE NAME; CHAR2 = TYPE;
915 - C                 CHAR3 = ACCESS; 12 = ERROR CODE (0=NONE, 1=ERROR)
916 - C
917 -         SUBROUTINE OCHECK(DEV,STRING,TYPE,AX,ER)
918 -         IMPLICIT REAL(A-H,O-Z)
919 -         INTEGER DEV,ER
920 -         CHARACTER STRING*36,TYPE*7,AX*10,RESP*1
921 -         IREP=0
922 - 10     CONTINUE
923 -         OPEN(UNIT=DEV,NAME=STRING,STATUS=TYPE,ACCESS=AX,FORM='FORMATTED',
924 - +      ERR=100)
925 -         ER=0
926 -         RETURN
927 - 100    CONTINUE
928 -         IF(TYPE(1:3),EQ,'OLD') THEN
929 -         PRINT *
930 - +      'ERROR - File is not in directory or has incorrect access.'
931 -         ELSE IF(TYPE(1:3),EQ,'NEW') THEN
932 -         PRINT *,'ERROR - File already exists in directory.'
933 -         PRINT *,'Do you wish to overwrite it?'
934 -         READ '(A1)',RESP
935 -         IF(RESP,EQ,'Y') THEN
936 -         OPEN(UNIT=DEV,NAME=STRING,STATUS='OLD',FORM='FORMATTED')
937 -         CLOSE(UNIT=DEV,STATUS='DELETE')
938 -         OPEN(UNIT=DEV,NAME=STRING,STATUS=TYPE,ACCESS=AX,
939 - +      FORM='FORMATTED',ERR=100)
940 -         RETURN
941 -         ELSE
942 -         END IF
943 -         ELSE
944 -         PRINT *,'ERROR - Unable to open file.'
945 -         END IF
946 -         IREP=IREP+1
947 -         IF(IREP,GT,3) THEN
948 -         PRINT *,'Sorry, unable to open file specified.'
949 -         ER=1
950 -         RETURN
951 -         ELSE
952 -         PRINT *,'Enter correct file name:'
953 -         READ '(A36)',STRING
954 -         GOTO 10
955 -         END IF
956 -         END

```


Appendix C

Listing of translation program

```

1 - C **** DRAW **** P. KALAB; VERSION 2.0; DECEMBER 4, 1983. ****
2 - C PROGRAM WHICH TRANSLATES DATA CREATED BY STARDYNE PREPROCESSOR
3 - C FOR COUPLED SHEAR WALLS INTO FORMAT COMPATIBLE WITH THE
4 - C PROGRAM "MOVIE", WHICH IS A GRAPHICS PACKAGE OPERATING ON
5 - C CARLETON UNIVERSITY'S CP-6 SYSTEM.
6 - C
7 - C * VARIABLES USED - ARRAYS *
8 - C
9 - C ICON(16) = temporary buffer which stores up to 16 numbers to be
10 - C      printed on one line
11 - C IA(50)  = JA node for beam element
12 - C IB(50)  = JB node for beam element
13 - C IC(50)  = JC node for beam element
14 - C ID(50)  = JD node for beam element
15 - C
16 - C * VARIABLES USED - CHARACTER *
17 - C
18 - C SFILE*36 = name of STARDYNE data file      (DEVICE = 1)
19 - C DFILE*36 = name of data file for MOVIE Input (DEVICE = 2)
20 - C LINE*72  = buffer which reads input data as character strings
21 - C WD*72    = buffer which stores info. displayed if EOF is reached
22 - C RESP*1   = location which reads and tests "Y" or "N" answers
23 - C
24 - C * VARIABLES USED - REAL *
25 - C
26 - C X,Y,Z    = nodal coordinates (followed by 1 or 2 because six values
27 - C      must be printed per line
28 - C
29 - C * VARIABLES USED - INTEGER *
30 - C
31 - C NJ       = number of joints
32 - C IQUAD    = number of quadrilateral elements
33 - C IBEAM    = number of beam elements
34 - C NCON     = number of element corners (IQUAD+IBeam)*4
35 - C NPT      = number of elements (IQUAD+IBeam)
36 - C IND      = index for ICON(16)
37 - C JA - JD  = corner nodes of elements
38 - C I,J      = general DO-LOOP counters
39 - C LOC      = statement no. preceding E.O.F. being read
40 - C
41 - C * SUBROUTINES CALLED *
42 - C
43 - C DCHECK    = used to check input errors when opening files
44 - C
45 - C
46 - C
47 - C      IMPLICIT REAL(A-H,O-Z)
48 - C      IMPLICIT INTEGER(I-N)
49 - C      CHARACTER SFILE*36,DFILE*36,RESP*1,LINE*72,WD*72
50 - C      DIMENSION IA(50),IB(50),IC(50),ID(50),ICON(16)
51 - C      1  FORMAT(A1)
52 - C      36  FORMAT(A36)
53 - C      72  FORMAT(A72)
54 - C      100 FORMAT(//
55 - C      + ' GRAPHICS TRANSLATOR FOR COUPLED SHEAR WALL DATA FILE' /
56 - C      + 1X,52'='//)
57 - C      WRITE(108,100)
58 - C
59 - C OPEN FILES AND MAKE FIRST PASS THROUGH DATA FILE TO DETERMINE
60 - C THE NUMBER OF NODES, ELEMENTS, AND BEAMS.
61 - C
62 - C 200 CONTINUE
63 - C PRINT *,

```

```

64 -      + 'Enter name of file where STARDYNE input data is stored:'
65 -      READ(105,36) SFILE
66 -      IF(SFILE(1:4).EQ.'STOP') PRINT *, '== ABORTED =='; STOP
67 -      CALL OCHECK(1,SFILE,'OLD','SEQUENTIAL',IER)
68 -      IF(IER.EQ.1) STOP
69 - 210 CONTINUE
70 -      PRINT *,
71 -      + 'Enter name of file where graphics data will be stored:'
72 -      READ(105,36) DFILE
73 -      IF(DFILE(1:4).EQ.'STOP') PRINT *, '== ABORTED =='; STOP
74 -      IF(DFILE.EQ.SFILE) PRINT *,
75 -      + 'ERROR - This is your source data file.'; GOTO 210
76 -      CALL OCHECK(2,DFILE,'NEW','SEQUENTIAL',IER)
77 -      IF(IER.EQ.1) STOP
78 -      PRINT *, 'Translating data. Please wait ...'
79 -      NJ=0
80 -      IQAD=0
81 -      IBEAM=0
82 -      READ(1,72,END=2000) LINE
83 -      LOC=210
84 -      WD=LINE
85 -      REPEAT 250, WHILE (LINE(1:4).NE.'NODE')
86 -        READ(1,72,END=2000) LINE
87 - 250 CONTINUE
88 -      LOC=250
89 -      WD=LINE
90 -      REPEAT 260, WHILE (LINE(1:4).EQ.'NODE')
91 -        NJ=NJ+1
92 -        READ(1,72,END=2000) LINE
93 - 260 CONTINUE
94 -      LOC=260
95 -      WD=LINE
96 -      REPEAT 270, WHILE (LINE(1:6).NE.'BPROP2')
97 -        READ(1,72,END=2000) LINE
98 - 270 CONTINUE
99 -      LOC=270
100 -      WD=LINE
101 -      REPEAT 280, WHILE (LINE(1:6).EQ.'BPROP2')
102 -        IBEAM=IBeam+1
103 -        READ(1,72,END=2000) LINE
104 - 280 CONTINUE
105 -      LOC=280
106 -      WD=LINE
107 -      REPEAT 290, WHILE (LINE(1:5).NE.'QUADB')
108 -        READ(1,72,END=2000) LINE
109 - 290 CONTINUE
110 -      LOC=290
111 -      WD=LINE
112 -      REPEAT 300, WHILE (LINE(1:5).EQ.'QUADB')
113 -        IQAD=IQAD+1
114 -        READ(1,72,END=2000) LINE
115 - 300 CONTINUE
116 -      NP=1
117 -      NPT=IQAD+IBeam
118 -      NCON=NPT*4
119 -      WRITE(2,310) NP,NJ,NPT,NCON
120 - 310 FORMAT(4I5)
121 -      WRITE(2,310) 1,NPT
122 - C
123 - C MAKE SECOND PASS AND EXTRACT NODE COORDINATES AND CONNECTIVITY.
124 - C WRITE 6 COORDINATES PER LINE AND 16 CORNER NODES PER LINE.
125 - C
126 -      REWIND 1
127 -      LOC=310
128 -      WD=LINE
129 -      REPEAT 320, WHILE (LINE(1:4).NE.'NODE')
130 -        READ(1,72,END=2000) LINE
131 - 320 CONTINUE
132 - 330 CONTINUE

```

```

133 -      READ(LINE,340) X1,Y1,Z1
134 - 340  FORMAT(8X,3G)
135 -      LOC=340
136 -      WD=LINE
137 -      READ(1,72,END=2000) LINE
138 -      IF(LINE(1:3).EQ.'END') THEN
139 -          WRITE(2,350) X1,Y1,Z1
140 - 350  FORMAT(6E12.5)
141 -      ELSE
142 -          READ(LINE,340) X2,Y2,Z2
143 -          WRITE(2,350) X1,Y1,Z1,X2,Y2,Z2
144 -          READ(1,72,END=2000) LINE
145 -          IF(LINE(1:3).NE.'END') GOTO 330
146 -      END IF
147 - C
148 - C READ CORNER NODES FOR BEAM ELEMENTS AND TREAT THESE AS SPECIAL
149 - C QUAD PLATES.
150 - C
151 -      LOC=350
152 -      WD=LINE
153 -      REPEAT 370, WHILE (LINE(1:5).NE.'BRECT')
154 -          READ(1,72,END=2000) LINE
155 - 370  CONTINUE
156 -      LOC=370
157 -      WD=LINE
158 -      DO 380 I=1,1BEAM-1
159 -          READ(1,72,END=2000) LINE
160 - 380  CONTINUE
161 -      LOC=380
162 -      WD=LINE
163 -      DO 450 I=1,1BEAM
164 -          READ(1,72,END=2000) LINE
165 -          IF(LINE(1:5).NE.'BRECT') PRINT *,
166 -          + 'ERROR - Missing expected BRECT line. == ABORTING =='; GOTO 800
167 -          READ(LINE(17:20),400) IA(I)
168 - 400  FORMAT(IG)
169 -          READ(1,72,END=2000) LINE
170 -          IF(LINE(1:5).NE.'BRECT') PRINT *,
171 -          + 'ERROR - Missing expected BRECT line. == ABORTING =='; GOTO 800
172 -          READ(LINE(25:28),400) ID(I)
173 - 450  CONTINUE
174 -      LOC=450
175 -      WD=LINE
176 -      DO 500 I=1,1BEAM
177 -          READ(1,72,END=2000) LINE
178 -          IF(LINE(1:5).NE.'BRECT') PRINT *,
179 -          + 'ERROR - Missing expected BRECT line. == ABORTING =='; GOTO 800
180 -          READ(LINE(17:20),400) IB(I)
181 -          READ(1,72,END=2000) LINE
182 -          IF(LINE(1:5).NE.'BRECT') PRINT *,
183 -          + 'ERROR - Missing expected BRECT line. == ABORTING =='; GOTO 800
184 -          READ(LINE(25:28),400) IC(I)
185 - 500  CONTINUE
186 - C
187 - C READ QUAD PLATE CONNECTIVITY AND PRINT 16 VALUES PER LINE.
188 - C
189 -      LOC=500
190 -      WD=LINE
191 -      REPEAT 510, WHILE (LINE(1:5).NE.'QUADB')
192 -          READ(1,72,END=2000) LINE
193 - 510  CONTINUE
194 -      IND=1
195 -      LOC=510
196 -      WD=LINE
197 - 550  CONTINUE
198 -          READ(LINE,560) JA,JB,JC,JD
199 - 560  FORMAT(12X,4G)
200 -          ICON(IND)=JA
201 -          IND=IND+1

```

```

202 -      IF(IND.EQ.17) WRITE(2,570) (ICON(J),J=1,16); IND=1
203 - 570  FORMAT(16I5)
204 -      ICON(IND)=JB
205 -      IND=IND+1
206 -      IF(IND.EQ.17) WRITE(2,570) (ICON(J),J=1,16); IND=1
207 -      ICON(IND)=JC
208 -      IND=IND+1
209 -      IF(IND.EQ.17) WRITE(2,570) (ICON(J),J=1,16); IND=1
210 -      ICON(IND)=JD
211 -      IND=IND+1
212 -      IF(IND.EQ.17) WRITE(2,570) (ICON(J),J=1,16); IND=1
213 -      READ(1,72,END=2000) LINE
214 -      IF(LINE(1:3).NE.'END') GOTO 550
215 -      DO 580 I=1, IBEAM
216 -          ICON(IND)=IA(I)
217 -          IND=IND+1
218 -          IF(IND.EQ.17) WRITE(2,570) (ICON(J),J=1,16); IND=1
219 -          ICON(IND)=IB(I)
220 -          IND=IND+1
221 -          IF(IND.EQ.17) WRITE(2,570) (ICON(J),J=1,16); IND=1
222 -          ICON(IND)=IC(I)
223 -          IND=IND+1
224 -          IF(IND.EQ.17) WRITE(2,570) (ICON(J),J=1,16); IND=1
225 -          ICON(IND)=ID(I)
226 -          IND=IND+1
227 -          IF(IND.EQ.17) WRITE(2,570) (ICON(J),J=1,16); IND=1
228 - 580  CONTINUE
229 -      IND=IND-1
230 -      IF((IND.GT.0).AND.(IND.LT.16))
231 -          + WRITE(2,570) (ICON(J),J=1,IND)
232 -      WRITE(108,590) DFILE
233 - 590  FORMAT(// 'Input data for MOVIE written on file ',A36//
234 -          + ' +++++ NORMAL TERMINATION +++++')
235 - 800  CONTINUE
236 -      CLOSE (UNIT=1,STATUS='KEEP')
237 -      CLOSE (UNIT=2,STATUS='KEEP')
238 -      PRINT *, 'Do you wish to translate another file (Y/N)?'
239 -      READ(105,1) RESP
240 -      IF(RESPEQ.'Y') THEN
241 -          GOTO 200
242 -      ELSE IF(RESPEQ.'N') THEN
243 -          STOP
244 -      ELSE
245 -          PRINT *, 'Please answer "Y" or "N":'
246 -          GOTO 800
247 -      END IF
248 - 2000 CONTINUE
249 -      WRITE(108,2010) LOC,WD
250 - 2010 FORMAT(// 'ERROR - End of file reached after statement',I4,
251 -          + ' of main program.')
252 -      + ' Incorrect file was opened or data may have been altered.'
253 -      + ' Error occurred after reading statement: '/A72)
254 -      GOTO 800
255 -      END
256 - C
257 - C **** OCHECK **** P. KALAB; VERSION 1.2; SEPTEMBER 19, 1983. ****
258 - C SUBROUTINE WHICH OPENS FILES AND CHECKS FOR ERROR CONDITIONS. THE
259 - C USER HAS 3 CHANCES TO CORRECT ANY MISTAKES HE MAY HAVE MADE.
260 - C * SYNTAX      OCHECK(I1,CHAR1*36,CHAR2*7,CHAR3*7,12)
261 - C * VARIABLES  * I1 = DEVICE NUMBER; CHAR1 = FILE NAME; CHAR2 = TYPE;
262 - C                CHAR3 = ACCESS; 12 = ERROR CODE (0=NONE, 1=ERROR)
263 - C
264 - C SUBROUTINE OCHECK(DEV,STRING,TYPE,AX,ER)
265 - C IMPLICIT REAL(A-H,O-Z)
266 - C INTEGER DEV,ER
267 - C CHARACTER STRING*36,TYPE*7,AX*10,RESP*1
268 - C IREP=0
269 - C 10  CONTINUE
270 - C OPEN(UNIT=DEV,NAME=STRING,STATUS=TYPE,ACCESS=AX,FORM='FORMATTED',

```

```

271 - + ERR=100)
272 - ER=0
273 - RETURN
274 - 100 CONTINUE
275 - IF(TYPE(1:3).EQ.'OLD') THEN
276 - PRINT *
277 - + 'ERROR - File is not in directory or has incorrect access.'
278 - ELSE IF(TYPE(1:3).EQ.'NEW') THEN
279 - PRINT *, 'ERROR - File already exists in directory.'
280 - PRINT *, 'Do you wish to overwrite it?'
281 - READ '(A1)', RESP
282 - IF(RESP.EQ.'Y') THEN
283 - OPEN(UNIT=DEV, NAME=STRING, STATUS='OLD', FORM='FORMATTED')
284 - CLOSE(UNIT=DEV, STATUS='DELETE')
285 - OPEN(UNIT=DEV, NAME=STRING, STATUS=TYPE, ACCESS='A',
286 - + FORM='FORMATTED', ERR=100)
287 - RETURN
288 - ELSE
289 - END IF
290 - ELSE
291 - PRINT *, 'ERROR - Unable to open file.'
292 - END IF
293 - IREP=IREP+1
294 - IF(IREP.GT.3) THEN
295 - PRINT *, 'Sorry, unable to open file specified.'
296 - ER=1
297 - RETURN
298 - ELSE
299 - PRINT *, 'Enter correct file name:'
300 - READ '(A36)', STRING
301 - GOTO 10
302 - END IF
303 - END

```

Appendix D

Dialogue with the computer during the execution of the preprocessor

STARDYNE PREPROCESSOR FOR COUPLED SHEAR WALLS - METRIC VERSION

Enter numerical data after each prompt in the appropriate units separated by spaces or commas.

Any of the following commands override normal execution:

STOP = Stops program immediately

GEOM = Geometry parameters are displayed and may be altered

LOAD = Load parameters are displayed and may be altered

Is this a "GEOMETRY ONLY" run?

?N

Enter name of file where Stardyne input will be stored:

?IN1

Enter name of file where postprocessor commands will be stored:

?CP1

Enter the following concrete parameters in proper units:

Density (kg/cu.m), Strength (MPa):

?2400,30

Enter the following widths in metres:

Wall 1, distance between walls, Wall 2:

?6,2,5.1

Enter the number of floors in building:

?6

For the first floor, enter the following data in metres:

Wall thickness, storey height, beam height:

?.2,4,1.5

Do storeys 2 to 6 have constant floor height, beam height and wall thickness?

?N

For each storey where the wall thickness, floor height, or beam height changes, enter the storey (2- 6) and the three parameters in metres ("0"). End with a blank line.

?2,.2,3,.5

?6,.2,3.5,1

?

GEOMETRY DATA

CONCRETE PARAMETERS:

1) Density = 2400. kg/cu.m; 2) Strength = 30.0 MPa; 3) E = 27691. MPa

WALL WIDTHS:

4) Wall 1 = 6.000 m; 5) Gap = 2.000 m; 6) Wall 2 = 5.100 m

7) Number of storeys = 6

8) No. of elements in horizontal direction in Wall 1 = 4

9) Maximum permissible aspect ratio = 10.0

STOREY	WALL THICKNESS (m)	STOREY HEIGHT (m)	BEAM HEIGHT (m)
1	.200	4.000	1.500
2	.200	3.000	.500
3	.200	3.000	.500
4	.200	3.000	.500
5	.200	3.000	.500
6	.200	3.500	1.000

Do you wish to make any changes (Y/N)?
?N

VERTICAL UDL - Positive value indicates downward force.
Enter the storey no. (1- 6) and the load magnitude, (kN/m)
where the load changes. End with a blank line.
?1,70
?

LATERAL LOAD - Positive value indicates force pushing on wall.
For linearly varying lateral load on Wall 1 (in.kN/m)
Enter magnitude at bottom, magnitude at top:
?

Shear walls are connected at 6 storeys.
Enter the storey no. and the point load in kN acting at top
of storey. End with a blank line.
?1,80
?2,150
?3,180
?4,240
?5,340
?6,380
?

APPLIED LOADS: *****

STOREY	VERTICAL UDL (kN/m)	LATERAL CONC. LOAD (kN)
1	70.000	80.000
2	70.000	150.000
3	70.000	180.000
4	70.000	240.000
5	70.000	340.000
6	70.000	380.000

LINEARLY VARYING LATERAL LOAD (kN/m):
7) Bottom = .000 ; 8) Top = .000

Do you wish to make any changes?
?N

Do you wish to change geometry data?
?N

Generating mesh. Please wait ...

Highest aspect ratio = 6.8
Max. allowable ratio = 10.0
No. of elements in horizontal direction in Wall 1 = 4
No. of elements in horizontal direction in Wall 2 = 3
No. of elements in vertical direction per storey = 4

If satisfied press RETURN, otherwise enter any character:
?

Processing geometry data. Please wait ...

No. of nodes: 225
No. of elastic beams: 6
No. of rigid beams: 24
No. of quad. plates: 168

If satisfied press RETURN, otherwise enter any character:
?

Processing static loads. Please wait ...

Geometry and load data for STARDYNE written on file IN1
Postprocessor commands written on file CP1

+++ NORMAL PROGRAM TERMINATION +++

Do you wish to process a new case (Y/N)?
?Y

Is this a "GEOMETRY ONLY" run?
?Y

Enter name of file where Stardyne input will be stored:
?IN1

ERROR - File already exists in directory.
Do you wish to overwrite it?
?N

Enter correct file name:
?*IN2

Enter the following concrete parameters in proper units:
Density (kg/cu.m), Strength (MPa):
?VERY DENSE AND VERY STRONG

Incorrect numerical value or too few arguments entered:
VERY DENSE AND VERY STRONG
Density (kg/cu.m), Strength (MPa):
?2400,THIRTY

Incorrect numerical value or too few arguments entered:
2400,THIRTY
Density (kg/cu.m), Strength (MPa):
?2400,25,REDUNDANT

Enter the following widths in metres:
Wall 1, distance between walls, Wall 2:
?GEOM

GEOMETRY DATA

CONCRETE PARAMETERS:
1) Density = 2400. kg/cu.m; 2) Strength = 25.0 MPa; 3) E = 25279. MPa

WALL WIDTHS:

4) Wall 1 = 6.000 m; 5) Gap = 2.000 m; 6) Wall 2 = 5.100 m

7) Number of storeys = 6

8) No. of elements in horizontal direction in Wall 1 = 4

9) Maximum permissible aspect ratio = 10.0

	WALL THICKNESS (m)	STOREY HEIGHT (m)	BEAM HEIGHT (m)
1	.200	4.000	1.500
2	.200	3.000	.500
3	.200	3.000	.500
4	.200	3.000	.500
5	.200	3.000	.500
6	.200	3.500	1.000

Do you wish to enter data from this level (Y/N)?

?Y

To change any parameter for code no. 1-9, enter the code number and the new parameter. End with a blank line.
?GEOM

You are already in geometry edit mode!
Proceed with normal entry of data:

?7,2

?4,3

?5,0

?6,4

?

CONCRETE PARAMETERS:

1) Density = 2400. kg/cu.m; 2) Strength = 25.0 MPa; 3) E = 25279. MPa

WALL WIDTHS:

4) Wall 1 = 3.000 m; 5) Gap = .000 m; 6) Wall 2 = 4.000 m

7) Number of storeys = 2

8) No. of elements in horizontal direction in Wall 1 = 4

9) Maximum permissible aspect ratio = 10.0

Do you wish to make any changes (Y/N)?

?N

	WALL THICKNESS (m)	STOREY HEIGHT (m)	BEAM HEIGHT (m)
1	.200	4.000	1.500
2	.200	3.000	.500

Do you wish to make any changes (Y/N)?

?Y

To change the wall thickness, storey height, or beam height, enter the storey number and all three values.
End with a blank line.

?2,.2,3,.1

?

	WALL THICKNESS (m)	STOREY HEIGHT (m)	BEAM HEIGHT (m)
1	.200	4.000	1.500
2	.200	3.000	.100

Do you wish to make any changes (Y/N)?
?N

ERROR - All widths must be greater than zero.
Enter the following widths in metres:
Wall 1, distance between walls, Wall 2:
?3,2,4

GEOMETRY DATA *****

CONCRETE PARAMETERS:

1) Density = 2400. kg/cu.m; 2) Strength = 25.0 MPa; 3) E = 25279. MPa

WALL WIDTHS:

4) Wall 1 = 3.000 m; 5) Gap = 2.000 m; 6) Wall 2 = 4.000 m

7) Number of storeys = 2

8) No. of elements in horizontal direction in Wall 1 = 4

9) Maximum permissible aspect ratio = 10.0

STOREY	WALL THICKNESS (m)	STOREY HEIGHT (m)	BEAM HEIGHT (m)
1	.200	4.000	1.500
2	.200	3.000	.100

Do you wish to make any changes (Y/N)?
?N

ERROR - Wall 1 must be wider than Wall 2. Swapping.

GEOMETRY DATA *****

CONCRETE PARAMETERS:

1) Density = 2400. kg/cu.m; 2) Strength = 25.0 MPa; 3) E = 25279. MPa

WALL WIDTHS:

4) Wall 1 = 4.000 m; 5) Gap = 2.000 m; 6) Wall 2 = 3.000 m

7) Number of storeys = 2

8) No. of elements in horizontal direction in Wall 1 = 4

9) Maximum permissible aspect ratio = 10.0

STOREY	WALL THICKNESS (m)	STOREY HEIGHT (m)	BEAM HEIGHT (m)
1	.200	4.000	1.500
2	.200	3.000	.100

Do you wish to make any changes (Y/N)?
?Y

To change any parameter for code no. 1-9, enter the code
number and the new parameter. End with a blank line.
?9,3
?

CONCRETE PARAMETERS:

1) Density = 2400. kg/cu.m; 2) Strength = 25.0 MPa; 3) E = 25279. MPa

WALL WIDTHS:

4) Wall 1 = 4.000 m; 5) Gap = 2.000 m; 6) Wall 2 = 3.000 m

- 7) Number of storeys = 2
 8) No. of elements in horizontal direction in Wall 1 = 4
 9) Maximum permissible aspect ratio = 3.0

Do you wish to make any changes (Y/N)?
 ?N

STOREY	WALL THICKNESS (m)	STOREY HEIGHT (m)	BEAM HEIGHT (m)
1	.200	4.000	1.500
2	.200	3.000	.100

Do you wish to make any changes (Y/N)?
 ?N

WARNING - Beam height ratio MAX/MIN appears too large: 15.0
 Mesh may be impossible to generate or may give wrong results.
 Do you wish to proceed (Y/N)?
 ?Y

Generating mesh. Please wait ...

Desired geometry exceeds limits of this program.
 Aspect ratio = 5.1 Beam height = 1.5 No. of hor. elem. in Wall 1 = 27
 Either beam is too deep or mesh is too fine.
 Desired geometry exceeds limits of this program.
 Aspect ratio = 5.0 Beam height = 1.5 No. of hor. elem. in Wall 2 = 20
 Either beam is too deep or mesh is too fine.
 Desired geometry exceeds limits of this program.
 Aspect ratio = 5.1 Beam height = 1.5 No. of hor. elem. in Wall 1 = 27
 Either beam is too deep or mesh is too fine.
 Desired geometry exceeds limits of this program.
 Aspect ratio = 5.0 Beam height = 1.5 No. of hor. elem. in Wall 2 = 20
 Either beam is too deep or mesh is too fine.

WARNING - Mesh which will be generated may produce incorrect results. Do you wish to proceed (Y/N)?
 ?Y

Highest aspect ratio = 3.0
 Max. allowable ratio = 3.0
 No. of elements in horizontal direction in Wall 1 = 27
 No. of elements in horizontal direction in Wall 2 = 20
 No. of elements in vertical direction per storey = 9

If satisfied press RETURN, otherwise enter any character:
 ?

Processing geometry data. Please wait ...

No. of nodes: 931
 No. of elastic beams: 2
 No. of rigid beams: 8
 No. of quad. plates: 846

If satisfied press RETURN, otherwise enter any character:
 ?OOPS

Enter "STOP" to abort, "GEOM" to edit geometry, or "GO" to continue:
 ?STOP

STOP

Appendix E

Input data file for the finite element program STARDYNE as created by the preprocessor

```

1 - START      0      9.81      6      0      02222222223
2 - 6-floor shear walls; Wall 1: 6.0 m; Gap: 2.0 m; Wall 2: 5.1 m
3 - MATLG      1 CONCRETE .27691E+08.15000E+00.23544E+02.00000E+00.00000E+00
4 - ENC
5 - NODE      1      .0000      .0000
6 - NODE      2      1.5000      .0000
7 - NODE      3      3.0000      .0000
8 - NODE      4      4.5000      .0000
9 - NODE      5      6.0000      .0000
10 - NODE     6      .0000      1.2500
11 - NODE     7      1.5000      1.2500
12 - NODE     8      3.0000      1.2500
13 - NODE     9      4.5000      1.2500
14 - NODE    10      6.0000      1.2500
15 - NODE    11      .0000      2.5000
16 - NODE    12      1.5000      2.5000
17 - NODE    13      3.0000      2.5000
18 - NODE    14      4.5000      2.5000
19 - NODE    15      6.0000      2.5000
20 - NODE    16      .0000      3.2500
21 - NODE    17      1.5000      3.2500
22 - NODE    18      3.0000      3.2500
23 - NODE    19      4.5000      3.2500
24 - NODE    20      6.0000      3.2500
25 - NODE    21      .0000      4.0000
26 - NODE    22      1.5000      4.0000
27 - NODE    23      3.0000      4.0000
28 - NODE    24      4.5000      4.0000
29 - NODE    25      6.0000      4.0000
30 - NODE    26      .0000      5.2500
31 - NODE    27      1.5000      5.2500
32 - NODE    28      3.0000      5.2500
33 - NODE    29      4.5000      5.2500
34 - NODE    30      6.0000      5.2500
35 - NODE    31      .0000      6.5000
36 - NODE    32      1.5000      6.5000
37 - NODE    33      3.0000      6.5000
38 - NODE    34      4.5000      6.5000
39 - NODE    35      6.0000      6.5000
40 - NODE    36      .0000      6.7500
41 - NODE    37      1.5000      6.7500
42 - NODE    38      3.0000      6.7500
43 - NODE    39      4.5000      6.7500
44 - NODE    40      6.0000      6.7500
45 - NODE    41      .0000      7.0000
46 - NODE    42      1.5000      7.0000
47 - NODE    43      3.0000      7.0000
48 - NODE    44      4.5000      7.0000
49 - NODE    45      6.0000      7.0000
50 - NODE    46      .0000      8.2500
51 - NODE    47      1.5000      8.2500
52 - NODE    48      3.0000      8.2500
53 - NODE    49      4.5000      8.2500
54 - NODE    50      6.0000      8.2500
55 - NODE    51      .0000      9.5000
56 - NODE    52      1.5000      9.5000
57 - NODE    53      3.0000      9.5000
58 - NODE    54      4.5000      9.5000
59 - NODE    55      6.0000      9.5000
60 - NODE    56      .0000      9.7500
61 - NODE    57      1.5000      9.7500
62 - NODE    58      3.0000      9.7500
63 - NODE    59      4.5000      9.7500
64 - NODE    60      6.0000      9.7500

```

65 - NODE 61	.0000	10.0000
66 - NODE 62	1.5000	10.0000
67 - NODE 63	3.0000	10.0000
68 - NODE 64	4.5000	10.0000
69 - NODE 65	6.0000	10.0000
70 - NODE 66	.0000	11.2500
71 - NODE 67	1.5000	11.2500
72 - NODE 68	3.0000	11.2500
73 - NODE 69	4.5000	11.2500
74 - NODE 70	6.0000	11.2500
75 - NODE 71	.0000	12.5000
76 - NODE 72	1.5000	12.5000
77 - NODE 73	3.0000	12.5000
78 - NODE 74	4.5000	12.5000
79 - NODE 75	6.0000	12.5000
80 - NODE 76	.0000	12.7500
81 - NODE 77	1.5000	12.7500
82 - NODE 78	3.0000	12.7500
83 - NODE 79	4.5000	12.7500
84 - NODE 80	6.0000	12.7500
85 - NODE 81	.0000	13.0000
86 - NODE 82	1.5000	13.0000
87 - NODE 83	3.0000	13.0000
88 - NODE 84	4.5000	13.0000
89 - NODE 85	6.0000	13.0000
90 - NODE 86	.0000	14.2500
91 - NODE 87	1.5000	14.2500
92 - NODE 88	3.0000	14.2500
93 - NODE 89	4.5000	14.2500
94 - NODE 90	6.0000	14.2500
95 - NODE 91	.0000	15.5000
96 - NODE 92	1.5000	15.5000
97 - NODE 93	3.0000	15.5000
98 - NODE 94	4.5000	15.5000
99 - NODE 95	6.0000	15.5000
100 - NODE 96	.0000	15.7500
101 - NODE 97	1.5000	15.7500
102 - NODE 98	3.0000	15.7500
103 - NODE 99	4.5000	15.7500
104 - NODE 100	6.0000	15.7500
105 - NODE 101	.0000	16.0000
106 - NODE 102	1.5000	16.0000
107 - NODE 103	3.0000	16.0000
108 - NODE 104	4.5000	16.0000
109 - NODE 105	6.0000	16.0000
110 - NODE 106	.0000	17.2500
111 - NODE 107	1.5000	17.2500
112 - NODE 108	3.0000	17.2500
113 - NODE 109	4.5000	17.2500
114 - NODE 110	6.0000	17.2500
115 - NODE 111	.0000	18.5000
116 - NODE 112	1.5000	18.5000
117 - NODE 113	3.0000	18.5000
118 - NODE 114	4.5000	18.5000
119 - NODE 115	6.0000	18.5000
120 - NODE 116	.0000	19.0000
121 - NODE 117	1.5000	19.0000
122 - NODE 118	3.0000	19.0000
123 - NODE 119	4.5000	19.0000
124 - NODE 120	6.0000	19.0000
125 - NODE 121	.0000	19.5000
126 - NODE 122	1.5000	19.5000
127 - NODE 123	3.0000	19.5000
128 - NODE 124	4.5000	19.5000
129 - NODE 125	6.0000	19.5000
130 - NODE 126	8.0000	.0000
131 - NODE 127	9.7000	.0000
132 - NODE 128	11.4000	.0000
133 - NODE 129	13.1000	.0000

134 - NODE 130	8.0000	1.2500
135 - NODE 131	9.7000	1.2500
136 - NODE 132	11.4000	1.2500
137 - NODE 133	13.1000	1.2500
138 - NODE 134	8.0000	2.5000
139 - NODE 135	9.7000	2.5000
140 - NODE 136	11.4000	2.5000
141 - NODE 137	13.1000	2.5000
142 - NODE 138	8.0000	3.2500
143 - NODE 139	9.7000	3.2500
144 - NODE 140	11.4000	3.2500
145 - NODE 141	13.1000	3.2500
146 - NODE 142	8.0000	4.0000
147 - NODE 143	9.7000	4.0000
148 - NODE 144	11.4000	4.0000
149 - NODE 145	13.1000	4.0000
150 - NODE 146	8.0000	5.2500
151 - NODE 147	9.7000	5.2500
152 - NODE 148	11.4000	5.2500
153 - NODE 149	13.1000	5.2500
154 - NODE 150	8.0000	6.5000
155 - NODE 151	9.7000	6.5000
156 - NODE 152	11.4000	6.5000
157 - NODE 153	13.1000	6.5000
158 - NODE 154	8.0000	6.7500
159 - NODE 155	9.7000	6.7500
160 - NODE 156	11.4000	6.7500
161 - NODE 157	13.1000	6.7500
162 - NODE 158	8.0000	7.0000
163 - NODE 159	9.7000	7.0000
164 - NODE 160	11.4000	7.0000
165 - NODE 161	13.1000	7.0000
166 - NODE 162	8.0000	8.2500
167 - NODE 163	9.7000	8.2500
168 - NODE 164	11.4000	8.2500
169 - NODE 165	13.1000	8.2500
170 - NODE 166	8.0000	9.5000
171 - NODE 167	9.7000	9.5000
172 - NODE 168	11.4000	9.5000
173 - NODE 169	13.1000	9.5000
174 - NODE 170	8.0000	9.7500
175 - NODE 171	9.7000	9.7500
176 - NODE 172	11.4000	9.7500
177 - NODE 173	13.1000	9.7500
178 - NODE 174	8.0000	10.0000
179 - NODE 175	9.7000	10.0000
180 - NODE 176	11.4000	10.0000
181 - NODE 177	13.1000	10.0000
182 - NODE 178	8.0000	11.2500
183 - NODE 179	9.7000	11.2500
184 - NODE 180	11.4000	11.2500
185 - NODE 181	13.1000	11.2500
186 - NODE 182	8.0000	12.5000
187 - NODE 183	9.7000	12.5000
188 - NODE 184	11.4000	12.5000
189 - NODE 185	13.1000	12.5000
190 - NODE 186	8.0000	12.7500
191 - NODE 187	9.7000	12.7500
192 - NODE 188	11.4000	12.7500
193 - NODE 189	13.1000	12.7500
194 - NODE 190	8.0000	13.0000
195 - NODE 191	9.7000	13.0000
196 - NODE 192	11.4000	13.0000
197 - NODE 193	13.1000	13.0000
198 - NODE 194	8.0000	14.2500
199 - NODE 195	9.7000	14.2500
200 - NODE 196	11.4000	14.2500
201 - NODE 197	13.1000	14.2500
202 - NODE 198	8.0000	15.5000

203 - NODE 199	9.7000	15.5000				
204 - NODE 200	11.4000	15.5000				
205 - NODE 201	13.1000	15.5000				
206 - NODE 202	8.0000	15.7500				
207 - NODE 203	9.7000	15.7500				
208 - NODE 204	11.4000	15.7500				
209 - NODE 205	13.1000	15.7500				
210 - NODE 206	8.0000	16.0000				
211 - NODE 207	9.7000	16.0000				
212 - NODE 208	11.4000	16.0000				
213 - NODE 209	13.1000	16.0000				
214 - NODE 210	8.0000	17.2500				
215 - NODE 211	9.7000	17.2500				
216 - NODE 212	11.4000	17.2500				
217 - NODE 213	13.1000	17.2500				
218 - NODE 214	8.0000	18.5000				
219 - NODE 215	9.7000	18.5000				
220 - NODE 216	11.4000	18.5000				
221 - NODE 217	13.1000	18.5000				
222 - NODE 218	8.0000	19.0000				
223 - NODE 219	9.7000	19.0000				
224 - NODE 220	11.4000	19.0000				
225 - NODE 221	13.1000	19.0000				
226 - NODE 222	8.0000	19.5000				
227 - NODE 223	9.7000	19.5000				
228 - NODE 224	11.4000	19.5000				
229 - NODE 225	13.1000	19.5000				
230 - END						
231 - RESTG	1 5	1123456				
232 - RESTG	126 129	1123456				
233 - END						
234 - BRECT	1	209999 138	0 25	1 1		
235 - BRECT	2	409999 154	0 45	1 2		
236 - BRECT	3	609999 170	0 65	1 3		
237 - BRECT	4	809999 186	0 85	1 4		
238 - BRECT	5	1009999 202	0 105	1 5		
239 - BRECT	6	1209999 218	0 125	1 6		
240 - BRECT	7	159999 20	0 14	1 7		
241 - BRECT	8	209999 25	0 19	1 7		
242 - BRECT	9	359999 40	0 34	1 7		
243 - BRECT	10	409999 45	0 39	1 7		
244 - BRECT	11	559999 60	0 54	1 7		
245 - BRECT	12	609999 65	0 59	1 7		
246 - BRECT	13	759999 80	0 74	1 7		
247 - BRECT	14	809999 85	0 79	1 7		
248 - BRECT	15	959999 100	0 94	1 7		
249 - BRECT	16	1009999 105	0 99	1 7		
250 - BRECT	17	1159999 120	0 114	1 7		
251 - BRECT	18	1209999 125	0 119	1 7		
252 - BRECT	19	1349999 138	0 15	1 7		
253 - BRECT	20	1389999 142	0 20	1 7		
254 - BRECT	21	1509999 154	0 35	1 7		
255 - BRECT	22	1549999 158	0 40	1 7		
256 - BRECT	23	1669999 170	0 55	1 7		
257 - BRECT	24	1709999 174	0 60	1 7		
258 - BRECT	25	1829999 186	0 75	1 7		
259 - BRECT	26	1869999 190	0 80	1 7		
260 - BRECT	27	1989999 202	0 95	1 7		
261 - BRECT	28	2029999 206	0 100	1 7		
262 - BRECT	29	2149999 218	0 115	1 7		
263 - BRECT	30	2189999 222	0 120	1 7		
264 - END						
265 - BPROP2	1	.20	1.50	1.50	1.50	.00
266 - BPROP2	2	.20	.50	1.50	1.50	.00
267 - BPROP2	3	.20	.50	1.50	1.50	.00
268 - BPROP2	4	.20	.50	1.50	1.50	.00
269 - BPROP2	5	.20	.50	1.50	1.50	.00
270 - BPROP2	6	.20	1.00	1.50	1.50	.00
271 - BPROP6	7	1				

272 - END									
273 - QUADB	1	1	2	7	6	1	2	.200	.0 2
274 - QUADB	2	2	3	8	7	1	2	.200	.0 2
275 - QUADB	3	3	4	9	8	1	2	.200	.0 2
276 - QUADB	4	4	5	10	9	1	2	.200	.0 2
277 - QUADB	5	6	7	12	11	1	2	.200	.0 2
278 - QUADB	6	7	8	13	12	1	2	.200	.0 2
279 - QUADB	7	8	9	14	13	1	2	.200	.0 2
280 - QUADB	8	9	10	15	14	1	2	.200	.0 2
281 - QUADB	9	11	12	17	16	1	2	.200	.0 2
282 - QUADB	10	12	13	18	17	1	2	.200	.0 2
283 - QUADB	11	13	14	19	18	1	2	.200	.0 2
284 - QUADB	12	14	15	20	19	1	2	.200	.0 2
285 - QUADB	13	16	17	22	21	1	2	.200	.0 2
286 - QUADB	14	17	18	23	22	1	2	.200	.0 2
287 - QUADB	15	18	19	24	23	1	2	.200	.0 2
288 - QUADB	16	19	20	25	24	1	2	.200	.0 2
289 - QUADB	17	21	22	27	26	1	2	.200	.0 2
290 - QUADB	18	22	23	28	27	1	2	.200	.0 2
291 - QUADB	19	23	24	29	28	1	2	.200	.0 2
292 - QUADB	20	24	25	30	29	1	2	.200	.0 2
293 - QUADB	21	26	27	32	31	1	2	.200	.0 2
294 - QUADB	22	27	28	33	32	1	2	.200	.0 2
295 - QUADB	23	28	29	34	33	1	2	.200	.0 2
296 - QUADB	24	29	30	35	34	1	2	.200	.0 2
297 - QUADB	25	31	32	37	36	1	2	.200	.0 2
298 - QUADB	26	32	33	38	37	1	2	.200	.0 2
299 - QUADB	27	33	34	39	38	1	2	.200	.0 2
300 - QUADB	28	34	35	40	39	1	2	.200	.0 2
301 - QUADB	29	36	37	42	41	1	2	.200	.0 2
302 - QUADB	30	37	38	43	42	1	2	.200	.0 2
303 - QUADB	31	38	39	44	43	1	2	.200	.0 2
304 - QUADB	32	39	40	45	44	1	2	.200	.0 2
305 - QUADB	33	41	42	47	46	1	2	.200	.0 2
306 - QUADB	34	42	43	48	47	1	2	.200	.0 2
307 - QUADB	35	43	44	49	48	1	2	.200	.0 2
308 - QUADB	36	44	45	50	49	1	2	.200	.0 2
309 - QUADB	37	46	47	52	51	1	2	.200	.0 2
310 - QUADB	38	47	48	53	52	1	2	.200	.0 2
311 - QUADB	39	48	49	54	53	1	2	.200	.0 2
312 - QUADB	40	49	50	55	54	1	2	.200	.0 2
313 - QUADB	41	51	52	57	56	1	2	.200	.0 2
314 - QUADB	42	52	53	58	57	1	2	.200	.0 2
315 - QUADB	43	53	54	59	58	1	2	.200	.0 2
316 - QUADB	44	54	55	60	59	1	2	.200	.0 2
317 - QUADB	45	56	57	62	61	1	2	.200	.0 2
318 - QUADB	46	57	58	63	62	1	2	.200	.0 2
319 - QUADB	47	58	59	64	63	1	2	.200	.0 2
320 - QUADB	48	59	60	65	64	1	2	.200	.0 2
321 - QUADB	49	61	62	67	66	1	2	.200	.0 2
322 - QUADB	50	62	63	68	67	1	2	.200	.0 2
323 - QUADB	51	63	64	69	68	1	2	.200	.0 2
324 - QUADB	52	64	65	70	69	1	2	.200	.0 2
325 - QUADB	53	66	67	72	71	1	2	.200	.0 2
326 - QUADB	54	67	68	73	72	1	2	.200	.0 2
327 - QUADB	55	68	69	74	73	1	2	.200	.0 2
328 - QUADB	56	69	70	75	74	1	2	.200	.0 2
329 - QUADB	57	71	72	77	76	1	2	.200	.0 2
330 - QUADB	58	72	73	78	77	1	2	.200	.0 2
331 - QUADB	59	73	74	79	78	1	2	.200	.0 2
332 - QUADB	60	74	75	80	79	1	2	.200	.0 2
333 - QUADB	61	76	77	82	81	1	2	.200	.0 2
334 - QUADB	62	77	78	83	82	1	2	.200	.0 2
335 - QUADB	63	78	79	84	83	1	2	.200	.0 2
336 - QUADB	64	79	80	85	84	1	2	.200	.0 2
337 - QUADB	65	81	82	87	86	1	2	.200	.0 2
338 - QUADB	66	82	83	88	87	1	2	.200	.0 2
339 - QUADB	67	83	84	89	88	1	2	.200	.0 2
340 - QUADB	68	84	85	90	89	1	2	.200	.0 2

341 - QUADB	69	86	87	92	91	1	2	.200	.0 2
342 - QUADB	70	87	88	93	92	1	2	.200	.0 2
343 - QUADB	71	88	89	94	93	1	2	.200	.0 2
344 - QUADB	72	89	90	95	94	1	2	.200	.0 2
345 - QUADB	73	91	92	97	96	1	2	.200	.0 2
346 - QUADB	74	92	93	98	97	1	2	.200	.0 2
347 - QUADB	75	93	94	99	98	1	2	.200	.0 2
348 - QUADB	76	94	95	100	99	1	2	.200	.0 2
349 - QUADB	77	96	97	102	101	1	2	.200	.0 2
350 - QUADB	78	97	98	103	102	1	2	.200	.0 2
351 - QUADB	79	98	99	104	103	1	2	.200	.0 2
352 - QUADB	80	99	100	105	104	1	2	.200	.0 2
353 - QUADB	81	101	102	107	106	1	2	.200	.0 2
354 - QUADB	82	102	103	108	107	1	2	.200	.0 2
355 - QUADB	83	103	104	109	108	1	2	.200	.0 2
356 - QUADB	84	104	105	110	109	1	2	.200	.0 2
357 - QUADB	85	106	107	112	111	1	2	.200	.0 2
358 - QUADB	86	107	108	113	112	1	2	.200	.0 2
359 - QUADB	87	108	109	114	113	1	2	.200	.0 2
360 - QUADB	88	109	110	115	114	1	2	.200	.0 2
361 - QUADB	89	111	112	117	116	1	2	.200	.0 2
362 - QUADB	90	112	113	118	117	1	2	.200	.0 2
363 - QUADB	91	113	114	119	118	1	2	.200	.0 2
364 - QUADB	92	114	115	120	119	1	2	.200	.0 2
365 - QUADB	93	116	117	122	121	1	2	.200	.0 2
366 - QUADB	94	117	118	123	122	1	2	.200	.0 2
367 - QUADB	95	118	119	124	123	1	2	.200	.0 2
368 - QUADB	96	119	120	125	124	1	2	.200	.0 2
369 - QUADB	97	126	127	131	130	1	2	.200	.0 2
370 - QUADB	98	127	128	132	131	1	2	.200	.0 2
371 - QUADB	99	128	129	133	132	1	2	.200	.0 2
372 - QUADB	100	130	131	135	134	1	2	.200	.0 2
373 - QUADB	101	131	132	136	135	1	2	.200	.0 2
374 - QUADB	102	132	133	137	136	1	2	.200	.0 2
375 - QUADB	103	134	135	139	138	1	2	.200	.0 2
376 - QUADB	104	135	136	140	139	1	2	.200	.0 2
377 - QUADB	105	136	137	141	140	1	2	.200	.0 2
378 - QUADB	106	138	139	143	142	1	2	.200	.0 2
379 - QUADB	107	139	140	144	143	1	2	.200	.0 2
380 - QUADB	108	140	141	145	144	1	2	.200	.0 2
381 - QUADB	109	142	143	147	146	1	2	.200	.0 2
382 - QUADB	110	143	144	148	147	1	2	.200	.0 2
383 - QUADB	111	144	145	149	148	1	2	.200	.0 2
384 - QUADB	112	146	147	151	150	1	2	.200	.0 2
385 - QUADB	113	147	148	152	151	1	2	.200	.0 2
386 - QUADB	114	148	149	153	152	1	2	.200	.0 2
387 - QUADB	115	150	151	155	154	1	2	.200	.0 2
388 - QUADB	116	151	152	156	155	1	2	.200	.0 2
389 - QUADB	117	152	153	157	156	1	2	.200	.0 2
390 - QUADB	118	154	155	159	158	1	2	.200	.0 2
391 - QUADB	119	155	156	160	159	1	2	.200	.0 2
392 - QUADB	120	156	157	161	160	1	2	.200	.0 2
393 - QUADB	121	158	159	163	162	1	2	.200	.0 2
394 - QUADB	122	159	160	164	163	1	2	.200	.0 2
395 - QUADB	123	160	161	165	164	1	2	.200	.0 2
396 - QUADB	124	162	163	167	166	1	2	.200	.0 2
397 - QUADB	125	163	164	168	167	1	2	.200	.0 2
398 - QUADB	126	164	165	169	168	1	2	.200	.0 2
399 - QUADB	127	166	167	171	170	1	2	.200	.0 2
400 - QUADB	128	167	168	172	171	1	2	.200	.0 2
401 - QUADB	129	168	169	173	172	1	2	.200	.0 2
402 - QUADB	130	170	171	175	174	1	2	.200	.0 2
403 - QUADB	131	171	172	176	175	1	2	.200	.0 2
404 - QUADB	132	172	173	177	176	1	2	.200	.0 2
405 - QUADB	133	174	175	179	178	1	2	.200	.0 2
406 - QUADB	134	175	176	180	179	1	2	.200	.0 2
407 - QUADB	135	176	177	181	180	1	2	.200	.0 2
408 - QUADB	136	178	179	183	182	1	2	.200	.0 2
409 - QUADB	137	179	180	184	183	1	2	.200	.0 2

410 - QUADB	138	180	181	185	184	1	2	.200	.0 2
411 - QUADB	139	182	183	187	186	1	2	.200	.0 2
412 - QUADB	140	183	184	188	187	1	2	.200	.0 2
413 - QUADB	141	184	185	189	188	1	2	.200	.0 2
414 - QUADB	142	186	187	191	190	1	2	.200	.0 2
415 - QUADB	143	187	188	192	191	1	2	.200	.0 2
416 - QUADB	144	188	189	193	192	1	2	.200	.0 2
417 - QUADB	145	190	191	195	194	1	2	.200	.0 2
418 - QUADB	146	191	192	196	195	1	2	.200	.0 2
419 - QUADB	147	192	193	197	196	1	2	.200	.0 2
420 - QUADB	148	194	195	199	198	1	2	.200	.0 2
421 - QUADB	149	195	196	200	199	1	2	.200	.0 2
422 - QUADB	150	196	197	201	200	1	2	.200	.0 2
423 - QUADB	151	198	199	203	202	1	2	.200	.0 2
424 - QUADB	152	199	200	204	203	1	2	.200	.0 2
425 - QUADB	153	200	201	205	204	1	2	.200	.0 2
426 - QUADB	154	202	203	207	206	1	2	.200	.0 2
427 - QUADB	155	203	204	208	207	1	2	.200	.0 2
428 - QUADB	156	204	205	209	208	1	2	.200	.0 2
429 - QUADB	157	206	207	211	210	1	2	.200	.0 2
430 - QUADB	158	207	208	212	211	1	2	.200	.0 2
431 - QUADB	159	208	209	213	212	1	2	.200	.0 2
432 - QUADB	160	210	211	215	214	1	2	.200	.0 2
433 - QUADB	161	211	212	216	215	1	2	.200	.0 2
434 - QUADB	162	212	213	217	216	1	2	.200	.0 2
435 - QUADB	163	214	215	219	218	1	2	.200	.0 2
436 - QUADB	164	215	216	220	219	1	2	.200	.0 2
437 - QUADB	165	216	217	221	220	1	2	.200	.0 2
438 - QUADB	166	218	219	223	222	1	2	.200	.0 2
439 - QUADB	167	219	220	224	223	1	2	.200	.0 2
440 - QUADB	168	220	221	225	224	1	2	.200	.0 2
441 - END									
442 - ENDGEOM									
443 - Static analysis with the following loads:									
444 - STATIC	0	0	0	1					
445 - Gravity, Vertical UDL on floors, Lateral LBL & point loads on Wall 1									
446 - BMLoad	1	1	OGL	.00	-70.00			.00	
447 - BMLoad	2	2	OGL	.00	-70.00			.00	
448 - BMLoad	3	3	OGL	.00	-70.00			.00	
449 - BMLoad	4	4	OGL	.00	-70.00			.00	
450 - BMLoad	5	5	OGL	.00	-70.00			.00	
451 - BMLoad	6	6	OGL	.00	-70.00			.00	
452 - END									
453 - CONC	1		.000	-2.207					
454 - CONC	2		.000	-4.414					
455 - CONC	3		.000	-4.414					
456 - CONC	4		.000	-4.414					
457 - CONC	5		.000	-2.207					
458 - CONC	6		.000	-4.414					
459 - CONC	7		.000	-8.829					
460 - CONC	8		.000	-8.829					
461 - CONC	9		.000	-8.829					
462 - CONC	10		.000	-4.414					
463 - CONC	11		.000	-3.532					
464 - CONC	12		.000	-7.063					
465 - CONC	13		.000	-7.063					
466 - CONC	14		.000	-7.063					
467 - CONC	15		.000	-3.532					
468 - CONC	16		.000	-2.649					
469 - CONC	17		.000	-5.297					
470 - CONC	18		.000	-5.297					
471 - CONC	19		.000	-5.297					
472 - CONC	20		.000	-2.649					
473 - CONC	21	80.000		-56.032					
474 - CONC	22		.000	-112.063					
475 - CONC	23		.000	-112.063					
476 - CONC	24		.000	-112.063					
477 - CONC	25		.000	-63.095					
478 - CONC	26		.000	-4.414					

479 - CONC	27	.000	-8.829
480 - CONC	28	.000	-8.829
481 - CONC	29	.000	-8.829
482 - CONC	30	.000	-4.414
483 - CONC	31	.000	-2.649
484 - CONC	32	.000	-5.297
485 - CONC	33	.000	-5.297
486 - CONC	34	.000	-5.297
487 - CONC	35	.000	-2.649
488 - CONC	36	.000	-.883
489 - CONC	37	.000	-1.766
490 - CONC	38	.000	-1.766
491 - CONC	39	.000	-1.766
492 - CONC	40	.000	-.883
493 - CONC	41	150.000	-55.149
494 - CONC	42	.000	-110.297
495 - CONC	43	.000	-110.297
496 - CONC	44	.000	-110.297
497 - CONC	45	.000	-57.503
498 - CONC	46	.000	-4.414
499 - CONC	47	.000	-8.829
500 - CONC	48	.000	-8.829
501 - CONC	49	.000	-8.829
502 - CONC	50	.000	-4.414
503 - CONC	51	.000	-2.649
504 - CONC	52	.000	-5.297
505 - CONC	53	.000	-5.297
506 - CONC	54	.000	-5.297
507 - CONC	55	.000	-2.649
508 - CONC	56	.000	-.883
509 - CONC	57	.000	-1.766
510 - CONC	58	.000	-1.766
511 - CONC	59	.000	-1.766
512 - CONC	60	.000	-.883
513 - CONC	61	180.000	-55.149
514 - CONC	62	.000	-110.297
515 - CONC	63	.000	-110.297
516 - CONC	64	.000	-110.297
517 - CONC	65	.000	-57.503
518 - CONC	66	.000	-4.414
519 - CONC	67	.000	-8.829
520 - CONC	68	.000	-8.829
521 - CONC	69	.000	-8.829
522 - CONC	70	.000	-4.414
523 - CONC	71	.000	-2.649
524 - CONC	72	.000	-5.297
525 - CONC	73	.000	-5.297
526 - CONC	74	.000	-5.297
527 - CONC	75	.000	-2.649
528 - CONC	76	.000	-.883
529 - CONC	77	.000	-1.766
530 - CONC	78	.000	-1.766
531 - CONC	79	.000	-1.766
532 - CONC	80	.000	-.883
533 - CONC	81	240.000	-55.149
534 - CONC	82	.000	-110.297
535 - CONC	83	.000	-110.297
536 - CONC	84	.000	-110.297
537 - CONC	85	.000	-57.503
538 - CONC	86	.000	-4.414
539 - CONC	87	.000	-8.829
540 - CONC	88	.000	-8.829
541 - CONC	89	.000	-8.829
542 - CONC	90	.000	-4.414
543 - CONC	91	.000	-2.649
544 - CONC	92	.000	-5.297
545 - CONC	93	.000	-5.297
546 - CONC	94	.000	-5.297
547 - CONC	95	.000	-2.649

548 - CONC	96	.000	-.883
549 - CONC	97	.000	-1.766
550 - CONC	98	.000	-1.766
551 - CONC	99	.000	-1.766
552 - CONC	100	.000	-.883
553 - CONC	101	340.000	-55.149
554 - CONC	102	.000	-110.297
555 - CONC	103	.000	-110.297
556 - CONC	104	.000	-110.297
557 - CONC	105	.000	-57.503
558 - CONC	106	.000	-4.414
559 - CONC	107	.000	-8.829
560 - CONC	108	.000	-8.829
561 - CONC	109	.000	-8.829
562 - CONC	110	.000	-4.414
563 - CONC	111	.000	-3.090
564 - CONC	112	.000	-6.180
565 - CONC	113	.000	-6.180
566 - CONC	114	.000	-6.180
567 - CONC	115	.000	-3.090
568 - CONC	116	.000	-1.766
569 - CONC	117	.000	-3.532
570 - CONC	118	.000	-3.532
571 - CONC	119	.000	-3.532
572 - CONC	120	.000	-1.766
573 - CONC	121	380.000	-53.383
574 - CONC	122	.000	-106.766
575 - CONC	123	.000	-106.766
576 - CONC	124	.000	-106.766
577 - CONC	125	.000	-58.092
578 - CONC	126	.000	-2.502
579 - CONC	127	.000	-5.003
580 - CONC	128	.000	-5.003
581 - CONC	129	.000	-2.502
582 - CONC	130	.000	-5.003
583 - CONC	131	.000	-10.006
584 - CONC	132	.000	-10.006
585 - CONC	133	.000	-5.003
586 - CONC	134	.000	-4.002
587 - CONC	135	.000	-8.005
588 - CONC	136	.000	-8.005
589 - CONC	137	.000	-4.002
590 - CONC	138	.000	-3.002
591 - CONC	139	.000	-6.004
592 - CONC	140	.000	-6.004
593 - CONC	141	.000	-3.002
594 - CONC	142	.000	-70.566
595 - CONC	143	.000	-127.005
596 - CONC	144	.000	-127.005
597 - CONC	145	.000	-63.502
598 - CONC	146	.000	-5.003
599 - CONC	147	.000	-10.006
600 - CONC	148	.000	-10.006
601 - CONC	149	.000	-5.003
602 - CONC	150	.000	-3.002
603 - CONC	151	.000	-6.004
604 - CONC	152	.000	-6.004
605 - CONC	153	.000	-3.002
606 - CONC	154	.000	-1.001
607 - CONC	155	.000	-2.001
608 - CONC	156	.000	-2.001
609 - CONC	157	.000	-1.001
610 - CONC	158	.000	-64.856
611 - CONC	159	.000	-125.004
612 - CONC	160	.000	-125.004
613 - CONC	161	.000	-62.502
614 - CONC	162	.000	-5.003
615 - CONC	163	.000	-10.006
616 - CONC	164	.000	-10.006

617 - CONC	165	.000	-5.003
618 - CONC	166	.000	-3.002
619 - CONC	167	.000	-6.004
620 - CONC	168	.000	-6.004
621 - CONC	169	.000	-3.002
622 - CONC	170	.000	-1.001
623 - CONC	171	.000	-2.001
624 - CONC	172	.000	-2.001
625 - CONC	173	.000	-1.001
626 - CONC	174	.000	-64.856
627 - CONC	175	.000	-125.004
628 - CONC	176	.000	-125.004
629 - CONC	177	.000	-62.502
630 - CONC	178	.000	-5.003
631 - CONC	179	.000	-10.006
632 - CONC	180	.000	-10.006
633 - CONC	181	.000	-5.003
634 - CONC	182	.000	-3.002
635 - CONC	183	.000	-6.004
636 - CONC	184	.000	-6.004
637 - CONC	185	.000	-3.002
638 - CONC	186	.000	-1.001
639 - CONC	187	.000	-2.001
640 - CONC	188	.000	-2.001
641 - CONC	189	.000	-1.001
642 - CONC	190	.000	-64.856
643 - CONC	191	.000	-125.004
644 - CONC	192	.000	-125.004
645 - CONC	193	.000	-62.502
646 - CONC	194	.000	-5.003
647 - CONC	195	.000	-10.006
648 - CONC	196	.000	-10.006
649 - CONC	197	.000	-5.003
650 - CONC	198	.000	-3.002
651 - CONC	199	.000	-6.004
652 - CONC	200	.000	-6.004
653 - CONC	201	.000	-3.002
654 - CONC	202	.000	-1.001
655 - CONC	203	.000	-2.001
656 - CONC	204	.000	-2.001
657 - CONC	205	.000	-1.001
658 - CONC	206	.000	-64.856
659 - CONC	207	.000	-125.004
660 - CONC	208	.000	-125.004
661 - CONC	209	.000	-62.502
662 - CONC	210	.000	-5.003
663 - CONC	211	.000	-10.006
664 - CONC	212	.000	-10.006
665 - CONC	213	.000	-5.003
666 - CONC	214	.000	-3.502
667 - CONC	215	.000	-7.004
668 - CONC	216	.000	-7.004
669 - CONC	217	.000	-3.502
670 - CONC	218	.000	-2.001
671 - CONC	219	.000	-4.002
672 - CONC	220	.000	-4.002
673 - CONC	221	.000	-2.001
674 - CONC	222	.000	-65.209
675 - CONC	223	.000	-121.001
676 - CONC	224	.000	-121.001
677 - CONC	225	.000	-60.501
678 - END			
679 - END CASE			
680 - ALL DONE			
681 - OUTPUT	1	222	222
682 - ENDMODEL			22

Appendix F

Selected sections of the STARDYNE output data file

**** POST-PROCESS FOR ELEMENT LOADS LINK ****

GRAVITY, VERTICAL		UDL ON FLOORS, LATERAL		LDL & POINT LOADS ON WALL 1		FOR OUTPUT VECTOR		1
NODE		NODAL	POINT	APPLIED	FORCES	X5	X6	
		X1	X2	X3	X4			
1	0.		-.220700E+01	0.	0.	0.	0.	
2	0.		-.441400E+01	0.	0.	0.	0.	
3	0.		-.441400E+01	0.	0.	0.	0.	
4	0.		-.441400E+01	0.	0.	0.	0.	
5	0.		-.220700E+01	0.	0.	0.	0.	
6	0.		-.441400E+01	0.	0.	0.	0.	
7	0.		-.882900E+01	0.	0.	0.	0.	
8	0.		-.882900E+01	0.	0.	0.	0.	
9	0.		-.882900E+01	0.	0.	0.	0.	
10	0.		-.441400E+01	0.	0.	0.	0.	
11	0.		-.353200E+01	0.	0.	0.	0.	
12	0.		-.706300E+01	0.	0.	0.	0.	
13	0.		-.706300E+01	0.	0.	0.	0.	
14	0.		-.706300E+01	0.	0.	0.	0.	
15	0.		-.353200E+01	0.	0.	0.	0.	
16	0.		-.264900E+01	0.	0.	0.	0.	
17	0.		-.529700E+01	0.	0.	0.	0.	
18	0.		-.529700E+01	0.	0.	0.	0.	
19	0.		-.529700E+01	0.	0.	0.	0.	
20	0.		-.726490E+02	0.	0.	0.	-.233333E+02	
21	0.	.800000E+02	-.560320E+02	0.	0.	0.	0.	
22	0.		-.112063E+03	0.	0.	0.	0.	
23	0.		-.112063E+03	0.	0.	0.	0.	
24	0.		-.112063E+03	0.	0.	0.	0.	
25	0.		-.630950E+02	0.	0.	0.	0.	
26	0.		-.441400E+01	0.	0.	0.	0.	
27	0.		-.882900E+01	0.	0.	0.	0.	
28	0.		-.882900E+01	0.	0.	0.	0.	
29	0.		-.882900E+01	0.	0.	0.	0.	
30	0.		-.441400E+01	0.	0.	0.	0.	
31	0.		-.264900E+01	0.	0.	0.	0.	
32	0.		-.529700E+01	0.	0.	0.	0.	
33	0.		-.529700E+01	0.	0.	0.	0.	
34	0.		-.529700E+01	0.	0.	0.	0.	
35	0.		-.264900E+01	0.	0.	0.	0.	
36	0.		-.883000E+00	0.	0.	0.	0.	
37	0.		-.176600E+01	0.	0.	0.	0.	
38	0.		-.176600E+01	0.	0.	0.	0.	
39	0.		-.176600E+01	0.	0.	0.	0.	
40	0.		-.708830E+02	0.	0.	0.	-.233333E+02	
41	0.	.150000E+03	-.551490E+02	0.	0.	0.	0.	
42	0.		-.110297E+03	0.	0.	0.	0.	
43	0.		-.110297E+03	0.	0.	0.	0.	
44	0.		-.110297E+03	0.	0.	0.	0.	
45	0.		-.575030E+02	0.	0.	0.	0.	
46	0.		-.441400E+01	0.	0.	0.	0.	
47	0.		-.882900E+01	0.	0.	0.	0.	
48	0.		-.882900E+01	0.	0.	0.	0.	
49	0.		-.882900E+01	0.	0.	0.	0.	
50	0.		-.441400E+01	0.	0.	0.	0.	
51	0.		-.264900E+01	0.	0.	0.	0.	
52	0.		-.529700E+01	0.	0.	0.	0.	
53	0.		-.529700E+01	0.	0.	0.	0.	
54	0.		-.529700E+01	0.	0.	0.	0.	
55	0.		-.264900E+01	0.	0.	0.	0.	
56	0.		-.883000E+00	0.	0.	0.	0.	
57	0.		-.176600E+01	0.	0.	0.	0.	
58	0.		-.176600E+01	0.	0.	0.	0.	
59	0.		-.176600E+01	0.	0.	0.	0.	

60	0.	-.708830E+02	0.	0.	0.	-.233333E+02
61	.180000E+03	-.551490E+02	0.	0.	0.	0.
62	0.	-.110297E+03	0.	0.	0.	0.
63	0.	-.110297E+03	0.	0.	0.	0.
64	0.	-.110297E+03	0.	0.	0.	0.
65	0.	-.575030E+02	0.	0.	0.	0.
66	0.	-.441400E+01	0.	0.	0.	0.
67	0.	-.882900E+01	0.	0.	0.	0.
68	0.	-.882900E+01	0.	0.	0.	0.
69	0.	-.882900E+01	0.	0.	0.	0.
70	0.	-.441400E+01	0.	0.	0.	0.
71	0.	-.264900E+01	0.	0.	0.	0.
72	0.	-.529700E+01	0.	0.	0.	0.
73	0.	-.529700E+01	0.	0.	0.	0.
74	0.	-.529700E+01	0.	0.	0.	0.
75	0.	-.264900E+01	0.	0.	0.	0.
76	0.	-.883000E+00	0.	0.	0.	0.
77	0.	-.176600E+01	0.	0.	0.	0.
78	0.	-.176600E+01	0.	0.	0.	0.
79	0.	-.176600E+01	0.	0.	0.	0.
80	0.	-.708830E+02	0.	0.	0.	-.233333E+02
81	.240000E+03	-.551490E+02	0.	0.	0.	0.
82	0.	-.110297E+03	0.	0.	0.	0.
83	0.	-.110297E+03	0.	0.	0.	0.
84	0.	-.110297E+03	0.	0.	0.	0.
85	0.	-.575030E+02	0.	0.	0.	0.
86	0.	-.441400E+01	0.	0.	0.	0.
87	0.	-.882900E+01	0.	0.	0.	0.
88	0.	-.882900E+01	0.	0.	0.	0.
89	0.	-.882900E+01	0.	0.	0.	0.
90	0.	-.441400E+01	0.	0.	0.	0.
91	0.	-.264900E+01	0.	0.	0.	0.
92	0.	-.529700E+01	0.	0.	0.	0.
93	0.	-.529700E+01	0.	0.	0.	0.
94	0.	-.529700E+01	0.	0.	0.	0.
95	0.	-.264900E+01	0.	0.	0.	0.
96	0.	-.883000E+00	0.	0.	0.	0.
97	0.	-.176600E+01	0.	0.	0.	0.
98	0.	-.176600E+01	0.	0.	0.	0.
99	0.	-.176600E+01	0.	0.	0.	0.
100	0.	-.708830E+02	0.	0.	0.	-.233333E+02
101	.340000E+03	-.551490E+02	0.	0.	0.	0.
102	0.	-.110297E+03	0.	0.	0.	0.
103	0.	-.110297E+03	0.	0.	0.	0.
104	0.	-.110297E+03	0.	0.	0.	0.
105	0.	-.575030E+02	0.	0.	0.	0.
106	0.	-.441400E+01	0.	0.	0.	0.
107	0.	-.882900E+01	0.	0.	0.	0.
108	0.	-.882900E+01	0.	0.	0.	0.
109	0.	-.882900E+01	0.	0.	0.	0.
110	0.	-.441400E+01	0.	0.	0.	0.
111	0.	-.309000E+01	0.	0.	0.	0.
112	0.	-.618000E+01	0.	0.	0.	0.
113	0.	-.618000E+01	0.	0.	0.	0.
114	0.	-.618000E+01	0.	0.	0.	0.
115	0.	-.309000E+01	0.	0.	0.	0.
116	0.	-.176600E+01	0.	0.	0.	0.
117	0.	-.353200E+01	0.	0.	0.	0.
118	0.	-.353200E+01	0.	0.	0.	0.
119	0.	-.353200E+01	0.	0.	0.	0.
120	0.	-.717660E+02	0.	0.	0.	-.233333E+02
121	.380000E+03	-.533830E+02	0.	0.	0.	0.
122	0.	-.106766E+03	0.	0.	0.	0.
123	0.	-.106766E+03	0.	0.	0.	0.
124	0.	-.106766E+03	0.	0.	0.	0.
125	0.	-.580920E+02	0.	0.	0.	0.
126	0.	-.250200E+01	0.	0.	0.	0.
127	0.	-.500300E+01	0.	0.	0.	0.
128	0.	-.500300E+01	0.	0.	0.	0.

129	0.	-.250200E+01	0.	0.	0.	0.
130	0.	-.500300E+01	0.	0.	0.	0.
131	0.	-.100060E+02	0.	0.	0.	0.
132	0.	-.100060E+02	0.	0.	0.	0.
133	0.	-.500300E+01	0.	0.	0.	0.
134	0.	-.400200E+01	0.	0.	0.	0.
135	0.	-.800500E+01	0.	0.	0.	0.
136	0.	-.800500E+01	0.	0.	0.	0.
137	0.	-.400200E+01	0.	0.	0.	0.
138	0.	-.730020E+02	0.	0.	0.	.233333E+02
139	0.	-.600400E+01	0.	0.	0.	0.
140	0.	-.600400E+01	0.	0.	0.	0.
141	0.	-.300200E+01	0.	0.	0.	0.
142	0.	-.705660E+02	0.	0.	0.	0.
143	0.	-.127005E+03	0.	0.	0.	0.
144	0.	-.127005E+03	0.	0.	0.	0.
145	0.	-.635020E+02	0.	0.	0.	0.
146	0.	-.500300E+01	0.	0.	0.	0.
147	0.	-.100060E+02	0.	0.	0.	0.
148	0.	-.100060E+02	0.	0.	0.	0.
149	0.	-.500300E+01	0.	0.	0.	0.
150	0.	-.300200E+01	0.	0.	0.	0.
151	0.	-.600400E+01	0.	0.	0.	0.
152	0.	-.600400E+01	0.	0.	0.	0.
153	0.	-.300200E+01	0.	0.	0.	0.
154	0.	-.710010E+02	0.	0.	0.	.233333E+02
155	0.	-.200100E+01	0.	0.	0.	0.
156	0.	-.200100E+01	0.	0.	0.	0.
157	0.	-.100100E+01	0.	0.	0.	0.
158	0.	-.648560E+02	0.	0.	0.	0.
159	0.	-.125004E+03	0.	0.	0.	0.
160	0.	-.125004E+03	0.	0.	0.	0.
161	0.	-.625020E+02	0.	0.	0.	0.
162	0.	-.500300E+01	0.	0.	0.	0.
163	0.	-.100060E+02	0.	0.	0.	0.
164	0.	-.100060E+02	0.	0.	0.	0.
165	0.	-.500300E+01	0.	0.	0.	0.
166	0.	-.300200E+01	0.	0.	0.	0.
167	0.	-.600400E+01	0.	0.	0.	0.
168	0.	-.600400E+01	0.	0.	0.	0.
169	0.	-.300200E+01	0.	0.	0.	0.
170	0.	-.710010E+02	0.	0.	0.	.233333E+02
171	0.	-.200100E+01	0.	0.	0.	0.
172	0.	-.200100E+01	0.	0.	0.	0.
173	0.	-.100100E+01	0.	0.	0.	0.
174	0.	-.648560E+02	0.	0.	0.	0.
175	0.	-.125004E+03	0.	0.	0.	0.
176	0.	-.125004E+03	0.	0.	0.	0.
177	0.	-.625020E+02	0.	0.	0.	0.
178	0.	-.500300E+01	0.	0.	0.	0.
179	0.	-.100060E+02	0.	0.	0.	0.
180	0.	-.100060E+02	0.	0.	0.	0.
181	0.	-.500300E+01	0.	0.	0.	0.
182	0.	-.300200E+01	0.	0.	0.	0.
183	0.	-.600400E+01	0.	0.	0.	0.
184	0.	-.600400E+01	0.	0.	0.	0.
185	0.	-.300200E+01	0.	0.	0.	0.
186	0.	-.710010E+02	0.	0.	0.	.233333E+02
187	0.	-.200100E+01	0.	0.	0.	0.
188	0.	-.200100E+01	0.	0.	0.	0.
189	0.	-.100100E+01	0.	0.	0.	0.
190	0.	-.648560E+02	0.	0.	0.	0.
191	0.	-.125004E+03	0.	0.	0.	0.
192	0.	-.125004E+03	0.	0.	0.	0.
193	0.	-.625020E+02	0.	0.	0.	0.
194	0.	-.500300E+01	0.	0.	0.	0.
195	0.	-.100060E+02	0.	0.	0.	0.
196	0.	-.100060E+02	0.	0.	0.	0.
197	0.	-.500300E+01	0.	0.	0.	0.

198	0.	-.300200E+01	0.	0.	0.	0.
199	0.	-.600400E+01	0.	0.	0.	0.
200	0.	-.600400E+01	0.	0.	0.	0.
201	0.	-.300200E+01	0.	0.	0.	0.
202	0.	-.710010E+02	0.	0.	0.	.233333E+02
203	0.	-.200100E+01	0.	0.	0.	0.
204	0.	-.200100E+01	0.	0.	0.	0.
205	0.	-.100100E+01	0.	0.	0.	0.
206	0.	-.648560E+02	0.	0.	0.	0.
207	0.	-.125004E+03	0.	0.	0.	0.
208	0.	-.125004E+03	0.	0.	0.	0.
209	0.	-.625020E+02	0.	0.	0.	0.
210	0.	-.500300E+01	0.	0.	0.	0.
211	0.	-.100060E+02	0.	0.	0.	0.
212	0.	-.100060E+02	0.	0.	0.	0.
213	0.	-.500300E+01	0.	0.	0.	0.
214	0.	-.350200E+01	0.	0.	0.	0.
215	0.	-.700400E+01	0.	0.	0.	0.
216	0.	-.700400E+01	0.	0.	0.	0.
217	0.	-.350200E+01	0.	0.	0.	0.
218	0.	-.720010E+02	0.	0.	0.	.233333E+02
219	0.	-.400200E+01	0.	0.	0.	0.
220	0.	-.400200E+01	0.	0.	0.	0.
221	0.	-.200100E+01	0.	0.	0.	0.
222	0.	-.652090E+02	0.	0.	0.	0.
223	0.	-.121001E+03	0.	0.	0.	0.
224	0.	-.121001E+03	0.	0.	0.	0.
225	0.	-.605010E+02	0.	0.	0.	0.

1

0

GRAVITY, VERTICAL UDL ON FLOORS, LATERAL UDL & POINT LOADS ON WALL 1

STATIC DEFLECTIONS FOR OUTPUT VECTOR 1

MAXIMUM ROTATION IS AT NODE 95 *DOF = 6 VALUE = -.638035E-03

MAXIMUM TRANSLATION IS AT NODE 121 *DOF = 1 VALUE = .108054E-01

***** TRANSLATIONS ***** ROTATIONS (RADIANS) *****

NODE	X1	X2	X3	X4	X5	X6
6	.1478757E-03	.2414684E-03	0.	0.	0.	0.
7	.1232837E-03	.4289564E-04	0.	0.	0.	0.
8	.1284893E-03	-.9575714E-04	0.	0.	0.	0.
9	.1242502E-03	-.2627969E-03	0.	0.	0.	0.
10	.1790852E-03	-.5223286E-03	0.	0.	0.	0.
11	.4156410E-03	.4364797E-03	0.	0.	0.	0.
12	.3944506E-03	.8645919E-04	0.	0.	0.	0.
13	.4048808E-03	-.2087784E-03	0.	0.	0.	0.
14	.4453884E-03	-.4747948E-03	0.	0.	0.	0.
15	.5062914E-03	-.1067549E-02	0.	0.	0.	-.2955035E-03
16	.6334375E-03	.5368867E-03	0.	0.	0.	0.
17	.6200624E-03	.1037352E-03	0.	0.	0.	0.
18	.6164767E-03	-.2705163E-03	0.	0.	0.	0.
19	.6969117E-03	-.6497299E-03	0.	0.	0.	0.
20	.7279191E-03	-.1067549E-02	0.	0.	0.	-.2955035E-03
21	.9078220E-03	.6230825E-03	0.	0.	0.	0.
22	.8718450E-03	.1177186E-03	0.	0.	0.	0.
23	.8824768E-03	-.3320587E-03	0.	0.	0.	0.
24	.9127435E-03	-.8131867E-03	0.	0.	0.	0.
25	.9495467E-03	-.1067549E-02	0.	0.	0.	-.2955035E-03
26	.1388179E-02	.7603438E-03	0.	0.	0.	0.
27	.1377282E-02	.1449302E-03	0.	0.	0.	0.
28	.1376964E-02	-.4244436E-03	0.	0.	0.	0.
29	.1368132E-02	-.9615335E-03	0.	0.	0.	0.
30	.1398307E-02	-.1536193E-02	0.	0.	0.	0.
31	.1986238E-02	.8721543E-03	0.	0.	0.	0.
32	.1965860E-02	.1556419E-03	0.	0.	0.	0.
33	.1954727E-02	-.5039548E-03	0.	0.	0.	0.
34	.1968269E-02	-.1135692E-02	0.	0.	0.	0.
35	.2015434E-02	-.1908556E-02	0.	0.	0.	-.4831204E-03
36	.2115213E-02	.8901368E-03	0.	0.	0.	0.
37	.2099230E-02	.1568142E-03	0.	0.	0.	0.
38	.2073143E-02	-.5173977E-03	0.	0.	0.	0.
39	.2106742E-02	-.1186515E-02	0.	0.	0.	0.

40	.2136214E-02	-.1908556E-02	0.	0.	0.	-.4831204E-03
41	.2270311E-02	.9059988E-03	0.	0.	0.	0.
42	.2215492E-02	.1588268E-03	0.	0.	0.	0.
43	.2212671E-02	-.5320552E-03	0.	0.	0.	0.
44	.2230937E-02	-.1235392E-02	0.	0.	0.	0.
45	.2256994E-02	-.1908556E-02	0.	0.	0.	-.4831204E-03
46	.2912674E-02	.9918074E-03	0.	0.	0.	0.
47	.2909059E-02	.1728306E-03	0.	0.	0.	0.
48	.2903347E-02	-.6064750E-03	0.	0.	0.	0.
49	.2906185E-02	-.1374482E-02	0.	0.	0.	0.
50	.2932353E-02	-.2191073E-02	0.	0.	0.	0.
51	.3671177E-02	.1057690E-02	0.	0.	0.	0.
52	.3654859E-02	.1654314E-03	0.	0.	0.	0.
53	.3647671E-02	-.6752823E-03	0.	0.	0.	0.
54	.3656290E-02	-.1505579E-02	0.	0.	0.	0.
55	.3692743E-02	-.2432040E-02	0.	0.	0.	-.5717158E-03
56	.3830632E-02	.1066288E-02	0.	0.	0.	0.
57	.3816355E-02	.1633038E-03	0.	0.	0.	0.
58	.3795102E-02	-.6870786E-03	0.	0.	0.	0.
59	.3820084E-02	-.1540917E-02	0.	0.	0.	0.
60	.3835672E-02	-.2432040E-02	0.	0.	0.	-.5717158E-03
61	.4015923E-02	.1072492E-02	0.	0.	0.	0.
62	.3959951E-02	.1624190E-03	0.	0.	0.	0.
63	.3959757E-02	-.6999977E-03	0.	0.	0.	0.
64	.3972841E-02	-.1574257E-02	0.	0.	0.	0.
65	.3978601E-02	-.2432040E-02	0.	0.	0.	-.5717158E-03
66	.4770392E-02	.1112782E-02	0.	0.	0.	0.
67	.4773088E-02	.1618870E-03	0.	0.	0.	0.
68	.4765091E-02	-.7577337E-03	0.	0.	0.	0.
69	.4763749E-02	-.1665118E-02	0.	0.	0.	0.
70	.4780629E-02	-.2611474E-02	0.	0.	0.	0.
71	.5624863E-02	.1137777E-02	0.	0.	0.	0.
72	.5610259E-02	.1422259E-03	0.	0.	0.	0.
73	.5598371E-02	-.8068936E-03	0.	0.	0.	0.
74	.5596986E-02	-.1751698E-02	0.	0.	0.	0.
75	.5618907E-02	-.2756600E-02	0.	0.	0.	-.6206214E-03
76	.5803176E-02	.1138381E-02	0.	0.	0.	0.
77	.5786050E-02	.1385571E-03	0.	0.	0.	0.
78	.5761014E-02	-.8156190E-03	0.	0.	0.	0.
79	.5773943E-02	-.1773768E-02	0.	0.	0.	0.
80	.5774062E-02	-.2756600E-02	0.	0.	0.	-.6206214E-03
81	.6009734E-02	.1136432E-02	0.	0.	0.	0.
82	.5941302E-02	.1367953E-03	0.	0.	0.	0.
83	.5938778E-02	-.8257243E-03	0.	0.	0.	0.
84	.5942172E-02	-.1793856E-02	0.	0.	0.	0.
85	.5929218E-02	-.2756600E-02	0.	0.	0.	-.6206214E-03
86	.6802749E-02	.1143939E-02	0.	0.	0.	0.
87	.6810314E-02	.1306821E-03	0.	0.	0.	0.
88	.6797100E-02	-.8649965E-03	0.	0.	0.	0.
89	.6790483E-02	-.1843667E-02	0.	0.	0.	0.
90	.6799994E-02	-.2851457E-02	0.	0.	0.	0.
91	.7696832E-02	.1148267E-02	0.	0.	0.	0.
92	.7680082E-02	.1062961E-03	0.	0.	0.	0.
93	.7663601E-02	-.8969287E-03	0.	0.	0.	0.
94	.7655437E-02	-.1893557E-02	0.	0.	0.	0.
95	.7661557E-02	-.2916376E-02	0.	0.	0.	-.6380348E-03
96	.7884414E-02	.1145465E-02	0.	0.	0.	0.
97	.7859842E-02	.1028193E-03	0.	0.	0.	0.
98	.7830227E-02	-.9035760E-03	0.	0.	0.	0.
99	.7836107E-02	-.1904704E-02	0.	0.	0.	0.
100	.7821065E-02	-.2916376E-02	0.	0.	0.	-.6380348E-03
101	.8107683E-02	.1140085E-02	0.	0.	0.	0.
102	.8013113E-02	.1022305E-03	0.	0.	0.	0.
103	.8012477E-02	-.9122444E-03	0.	0.	0.	0.
104	.8009300E-02	-.1913966E-02	0.	0.	0.	0.
105	.7980574E-02	-.2916376E-02	0.	0.	0.	-.6380348E-03
106	.8882112E-02	.1149262E-02	0.	0.	0.	0.
107	.8895008E-02	.9821881E-04	0.	0.	0.	0.
108	.8876662E-02	-.9448121E-03	0.	0.	0.	0.

109	.8869037E-02	-.1934332E-02	0.	0.	0.	0.
110	.8869132E-02	-.2930092E-02	0.	0.	0.	0.
111	.9805612E-02	.1199430E-02	0.	0.	0.	0.
112	.9783636E-02	.5673693E-04	0.	0.	0.	0.
113	.9747918E-02	-.9616271E-03	0.	0.	0.	0.
114	.9694037E-02	-.1968135E-02	0.	0.	0.	0.
115	.9691010E-02	-.2905661E-02	0.	0.	0.	-.5333952E-03
116	.1020185E-01	.1223741E-02	0.	0.	0.	0.
117	.1018134E-01	.3517139E-04	0.	0.	0.	0.
118	.1007200E-01	-.9608831E-03	0.	0.	0.	0.
119	.1002789E-01	-.1978436E-02	0.	0.	0.	0.
120	.9957708E-02	-.2905661E-02	0.	0.	0.	-.5333952E-03
121	.1080542E-01	.1242697E-02	0.	0.	0.	0.
122	.1047222E-01	.2489077E-04	0.	0.	0.	0.
123	.1043353E-01	-.9605643E-03	0.	0.	0.	0.
124	.1035184E-01	-.1983555E-02	0.	0.	0.	0.
125	.1022441E-01	-.2905661E-02	0.	0.	0.	-.5333952E-03
130	.1426311E-03	.1678561E-03	0.	0.	0.	0.
131	.1340352E-03	-.6761268E-04	0.	0.	0.	0.
132	.1468148E-03	-.2311361E-03	0.	0.	0.	0.
133	.2122484E-03	-.4717890E-03	0.	0.	0.	0.
134	.4619862E-03	.2869036E-03	0.	0.	0.	-.2943035E-03
135	.4240909E-03	-.1354486E-03	0.	0.	0.	0.
136	.4397378E-03	-.4699227E-03	0.	0.	0.	0.
137	.4980053E-03	-.8752029E-03	0.	0.	0.	0.
138	.6827138E-03	.2869036E-03	0.	0.	0.	-.2943035E-03
139	.6525208E-03	-.1677840E-03	0.	0.	0.	0.
140	.6629310E-03	-.6072861E-03	0.	0.	0.	0.
141	.7222915E-03	-.1096894E-02	0.	0.	0.	0.
142	.9034415E-03	.2869036E-03	0.	0.	0.	-.2943035E-03
143	.9054522E-03	-.2879070E-03	0.	0.	0.	0.
144	.9257492E-03	-.7381697E-03	0.	0.	0.	0.
145	.9752849E-03	-.1306338E-02	0.	0.	0.	0.
146	.1391746E-02	.3826456E-03	0.	0.	0.	0.
147	.1400052E-02	-.2895970E-03	0.	0.	0.	0.
148	.1424620E-02	-.9252401E-03	0.	0.	0.	0.
149	.1473636E-02	-.1617259E-02	0.	0.	0.	0.
150	.2007605E-02	.7158746E-03	0.	0.	0.	-.4077381E-03
151	.1974521E-02	-.3561088E-03	0.	0.	0.	0.
152	.2005562E-02	-.1106684E-02	0.	0.	0.	0.
153	.2049055E-02	-.1896133E-02	0.	0.	0.	0.
154	.2109539E-02	.4158746E-03	0.	0.	0.	-.4077381E-03
155	.2106431E-02	-.3684979E-03	0.	0.	0.	0.
156	.2129010E-02	-.1141711E-02	0.	0.	0.	0.
157	.2172844E-02	-.1948478E-02	0.	0.	0.	0.
158	.2211474E-02	.4158746E-03	0.	0.	0.	-.4077381E-03
159	.2240686E-02	-.3828482E-03	0.	0.	0.	0.
160	.2256214E-02	-.1175267E-02	0.	0.	0.	0.
161	.2297982E-02	-.2000292E-02	0.	0.	0.	0.
162	.2906761E-02	.4565923E-03	0.	0.	0.	0.
163	.2913594E-02	-.4529619E-03	0.	0.	0.	0.
164	.2931410E-02	-.1316289E-02	0.	0.	0.	0.
165	.2967003E-02	-.2229272E-02	0.	0.	0.	0.
166	.3673104E-02	.4421622E-03	0.	0.	0.	-.4957847E-03
167	.3636187E-02	-.3081746E-03	0.	0.	0.	0.
168	.3660522E-02	-.1452745E-02	0.	0.	0.	0.
169	.3690579E-02	-.2426635E-02	0.	0.	0.	0.
170	.3797050E-02	.4421622E-03	0.	0.	0.	-.4957847E-03
171	.3794062E-02	-.5207254E-03	0.	0.	0.	0.
172	.3810979E-02	-.1478415E-02	0.	0.	0.	0.
173	.3839903E-02	-.2462954E-02	0.	0.	0.	0.
174	.3920996E-02	.4421622E-03	0.	0.	0.	-.4957847E-03
175	.3954332E-02	-.5353891E-03	0.	0.	0.	0.
176	.3962643E-02	-.1502628E-02	0.	0.	0.	0.
177	.3990866E-02	-.2498814E-02	0.	0.	0.	0.
178	.4727684E-02	.4484081E-03	0.	0.	0.	0.
179	.4736041E-02	-.5974321E-03	0.	0.	0.	0.
180	.4746278E-02	-.1600209E-02	0.	0.	0.	0.
181	.4769503E-02	-.2648543E-02	0.	0.	0.	0.

182	.5580399E-02	.402678Z-03	0.	0.	0.	0.	-.5398904E-03
183	.5538484E-02	-.6423231E-03	0.	0.	0.	0.	0.
184	.5557598E-02	-.1695306E-02	0.	0.	0.	0.	0.
185	.5575285E-02	-.2769506E-02	0.	0.	0.	0.	0.
186	.5715371E-02	.402678Z-03	0.	0.	0.	0.	-.5398904E-03
187	.5709803E-02	-.6540075E-03	0.	0.	0.	0.	0.
188	.5721415E-02	-.1712585E-02	0.	0.	0.	0.	0.
189	.5737803E-02	-.2791118E-02	0.	0.	0.	0.	0.
190	.5850344E-02	.402678Z-03	0.	0.	0.	0.	-.5398904E-03
191	.5882813E-02	-.6678027E-03	0.	0.	0.	0.	0.
192	.5885069E-02	-.172844Z-02	0.	0.	0.	0.	0.
193	.5901354E-02	-.2812457E-02	0.	0.	0.	0.	0.
194	.6704227E-02	.3870394E-03	0.	0.	0.	0.	0.
195	.6713275E-02	-.7188808E-03	0.	0.	0.	0.	0.
196	.6715491E-02	-.1786468E-02	0.	0.	0.	0.	0.
197	.6727533E-02	-.2891961E-02	0.	0.	0.	0.	0.
198	.7591327E-02	.3242569E-03	0.	0.	0.	0.	-.5486502E-03
199	.7540445E-02	-.7501948E-03	0.	0.	0.	0.	0.
200	.7551649E-02	-.1843709E-02	0.	0.	0.	0.	0.
201	.7557839E-02	-.2948308E-02	0.	0.	0.	0.	0.
202	.7728490E-02	.3242569E-03	0.	0.	0.	0.	-.5486502E-03
203	.7714846E-02	-.7593564E-03	0.	0.	0.	0.	0.
204	.7717533E-02	-.1853511E-02	0.	0.	0.	0.	0.
205	.7722933E-02	-.2957828E-02	0.	0.	0.	0.	0.
206	.7865652E-02	.3242569E-03	0.	0.	0.	0.	-.5486502E-03
207	.7889773E-02	-.7704662E-03	0.	0.	0.	0.	0.
208	.7882730E-02	-.1861956E-02	0.	0.	0.	0.	0.
209	.7888023E-02	-.2967404E-02	0.	0.	0.	0.	0.
210	.8715615E-02	.2992546E-03	0.	0.	0.	0.	0.
211	.8720610E-02	-.8034164E-03	0.	0.	0.	0.	0.
212	.8710540E-02	-.1883728E-02	0.	0.	0.	0.	0.
213	.8711424E-02	-.2994483E-02	0.	0.	0.	0.	0.
214	.9591647E-02	.2261384E-03	0.	0.	0.	0.	-.5565591E-03
215	.9530437E-02	-.8074405E-03	0.	0.	0.	0.	0.
216	.9528291E-02	-.1906901E-02	0.	0.	0.	0.	0.
217	.9531224E-02	-.3011574E-02	0.	0.	0.	0.	0.
218	.9869927E-02	.2261384E-03	0.	0.	0.	0.	-.5565591E-03
219	.9856464E-02	-.8124975E-03	0.	0.	0.	0.	0.
220	.9854156E-02	-.1914059E-02	0.	0.	0.	0.	0.
221	.9856142E-02	-.3018054E-02	0.	0.	0.	0.	0.
222	.1014821E-01	.2261384E-03	0.	0.	0.	0.	-.5565591E-03
223	.1018107E-01	-.8201479E-03	0.	0.	0.	0.	0.
224	.1017876E-01	-.1919966E-02	0.	0.	0.	0.	0.
225	.1018147E-01	-.3024710E-02	0.	0.	0.	0.	0.

GRAVITY, VERTICAL UDL ON FLOORS, LATERAL UDL & POINT LOADS ON WALL 1							
				BEAM ELEMENT LOADS		FOR OUTPUT	VECTOR
				AXIAL	SHEAR	SHEAR	TORSION
				P	V2	V3	MT
BEAM NO	LOAD-LOCATION, NODE-PERCENT.						
0 1	20 JA 0,0	1.87767E+02	-8.63009E+00	0.	0.	0.	0.
1	1 50,0	-1.87767E+02	7.86301E+01	0.	0.	0.	0.
1	1 50,0	1.87767E+02	-7.86301E+01	0.	0.	0.	0.
1	138 JB 100,0	-1.87767E+02	1.48630E+02	0.	0.	0.	0.
0 2	40 JA 0,0	3.69331E+01	2.66556E+01	0.	0.	0.	0.
2	1 50,0	-3.69331E+01	4.33444E+01	0.	0.	0.	0.
2	1 50,0	3.69331E+01	-4.33444E+01	0.	0.	0.	0.
2	154 JB 100,0	-3.69331E+01	1.13344E+02	0.	0.	0.	0.
0 3	60 JA 0,0	5.34742E+01	1.68630E+01	0.	0.	0.	0.
3	1 50,0	-5.34742E+01	5.31370E+01	0.	0.	0.	0.
3	1 50,0	5.34742E+01	-5.31370E+01	0.	0.	0.	0.
3	170 JB 100,0	-5.34742E+01	1.23137E+02	0.	0.	0.	0.
0 4	80 JA 0,0	8.12609E+01	1.17661E+01	0.	0.	0.	0.
4	1 50,0	-8.12609E+01	5.82339E+01	0.	0.	0.	0.
4	1 50,0	8.12609E+01	-5.82339E+01	0.	0.	0.	0.
4	186 JB 100,0	-8.12609E+01	1.28234E+02	0.	0.	0.	0.
0 5	100 JA 0,0	1.28176E+02	1.03165E+01	0.	0.	0.	0.
5	1 50,0	-1.28176E+02	5.96835E+01	0.	0.	0.	0.
5	1 50,0	1.28176E+02	-5.96835E+01	0.	0.	0.	0.

-5.53134E+01 BRECT
 1.16833E+01
 -1.16833E+01
 -1.01947E+02
 -2.03590E+01 BRECT
 1.20146E+01
 -1.20146E+01
 -6.63299E+01
 -3.01541E+01 BRECT
 1.20171E+01
 -1.20171E+01
 -7.61199E+01
 -3.52732E+01 BRECT
 1.20393E+01
 -1.20393E+01
 -8.11947E+01
 -3.67627E+01 BRECT
 1.20792E+01
 -1.20792E+01

5	202	JB	100.0	-1.28176E+02	1.29683E+02	0.	0.	0.	-8.26043E+01	BRECT
0	6	120	JA	0.0	2.43074E+02	-4.38246E+01	0.	0.	-9.02775E+01	
6	1	50.0		-2.43074E+02	1.13825E+02	0.	0.	0.	1.14529E+01	
6	1	50.0		2.43074E+02	-1.13825E+02	0.	0.	0.	-1.14529E+01	
6	218	JB	100.0	-2.43074E+02	1.83825E+02	0.	0.	0.	-1.37372E+02	

1
0 GRAVITY, VERTICAL UDL ON FLOORS, LATERAL LDL & POINT LOADS ON WALL 1
0 MAXIMUM LOAD SUMMARY FOR

	AXIAL P	BEAM SHEAR V2	ELEMENT SHEAR V3	LOADS TORSION MT	FOR OUTPUT BENDING M2	VECTOR BENDING M3	1
0	MAXIMUM BEAM LOADS =	-3.69331E+01	1.83825E+02	0.	0.	1.20792E+01	
0	BEAM NOS. =	2.	6.	1.	1.	5.	
0	MINIMUM BEAM LOADS =	-2.43074E+02	-1.13825E+02	0.	0.	-1.37372E+02	
0	BEAM NOS. =	6.	6.	1.	1.	6.	

1
0 GRAVITY, VERTICAL UDL ON FLOORS, LATERAL LDL & POINT LOADS ON WALL 1

BEAM	—NODES—	BEAM FX1	END FX2	LOADS FX3	IN GLOBAL SYSTEM MX1	FOR OUTPUT MX2	VECTOR MX3	1
0	1	JA 20	.188E+03	-.863E+01	0.	0.	-.553E+02	
		JE 138	-.188E+03	.149E+03	0.	0.	-.102E+03	
0	2	JA 40	.369E+02	.267E+02	0.	0.	-.204E+02	
		JB 154	-.369E+02	.113E+03	0.	0.	-.663E+02	
0	3	JA 60	.535E+02	.169E+02	0.	0.	-.302E+02	
		JB 170	-.535E+02	.123E+03	0.	0.	-.761E+02	
0	4	JA 80	.813E+02	.118E+02	0.	0.	-.353E+02	
		JB 186	-.813E+02	.128E+03	0.	0.	-.812E+02	
0	5	JA 100	.128E+03	.103E+02	0.	0.	-.366E+02	
		JB 202	-.128E+03	.130E+03	0.	0.	-.826E+02	
0	6	JA 120	.243E+03	-.438E+02	0.	0.	-.903E+02	
		JB 218	-.243E+03	.184E+03	0.	0.	-.137E+03	

1
0 GRAVITY, VERTICAL UDL ON FLOORS, LATERAL LDL & POINT LOADS ON WALL 1

BEAM NO	BEAM LOAD-LOCATION, NODE—PERCENT	— TRANSVERSE — SHEAR (V2/K3*A)	SHEAR (V3/K2*A)	TORSIONAL (T*Q/J)	BEAM STRESSES — STRESSES AT USER LOCATIONS — A B C	FOR OUTPUT MAXIMUM (P/A+MC/1)	VECTOR MINIMUM (P/A-MC/1)	1 COMBINED SHEAR
1	20 JA	0.0	-3.38E+01	0.	-6.259E+02 -6.157E+03 -6.259E+02	4.9054E+03	-6.1572E+03	3.08E+03 BRECT
1	1	50.0	-3.08E+02	0.	-6.259E+02 -1.794E+03 -6.259E+02	5.4244E+02	-1.7942E+03	8.97E+02
1	1	50.0	-3.08E+02	0.	-6.259E+02 -1.794E+03 -6.259E+02	5.4244E+02	-1.7942E+03	8.97E+02
1	138 JB	100.0	-5.83E+02	0.	-6.259E+02 9.569E+03 -6.259E+02	9.5688E+03	-1.0821E+04	5.41E+03
2	40 JA	0.0	3.14E+02	0.	-3.693E+02 -6.477E+03 -3.693E+02	5.7384E+03	-6.4770E+03	3.24E+03 BRECT
2	1	50.0	-5.10E+02	0.	-3.693E+02 -3.974E+03 -3.693E+02	3.2350E+03	-3.9737E+03	1.99E+03
2	1	50.0	-5.10E+02	0.	-3.693E+02 -3.974E+03 -3.693E+02	3.2350E+03	-3.9737E+03	1.99E+03
2	154 JB	100.0	-1.33E+03	0.	-3.693E+02 1.953E+04 -3.693E+02	1.9530E+04	-2.0268E+04	1.01E+04
3	60 JA	0.0	1.98E+02	0.	-5.347E+02 -9.581E+03 -5.347E+02	8.5115E+03	-9.5810E+03	4.79E+03 BRECT
3	1	50.0	-6.25E+02	0.	-5.347E+02 -4.140E+03 -5.347E+02	3.0704E+03	-4.1399E+03	2.07E+03
3	1	50.0	-6.25E+02	0.	-5.347E+02 -4.140E+03 -5.347E+02	3.0704E+03	-4.1399E+03	2.07E+03
3	170 JB	100.0	-1.45E+03	0.	-5.347E+02 2.230E+04 -5.347E+02	2.2301E+04	-2.3371E+04	1.17E+04
4	80 JA	0.0	1.38E+02	0.	-8.126E+02 -1.139E+04 -8.126E+02	9.7693E+03	-1.1395E+04	5.70E+03 BRECT
4	1	50.0	-6.85E+02	0.	-8.126E+02 -4.424E+03 -8.126E+02	2.7992E+03	-4.4244E+03	2.21E+03
4	1	50.0	-6.85E+02	0.	-8.126E+02 -4.424E+03 -8.126E+02	2.7992E+03	-4.4244E+03	2.21E+03
4	186 JB	100.0	-1.51E+03	0.	-8.126E+02 2.355E+04 -8.126E+02	2.3546E+04	-2.5171E+04	1.26E+04
5	100 JA	0.0	1.21E+02	0.	-1.282E+03 -1.231E+04 -1.282E+03	9.7470E+03	-1.2311E+04	6.16E+03 BRECT
5	1	50.0	-7.02E+02	0.	-1.282E+03 -4.906E+03 -1.282E+03	2.3420E+03	-4.9055E+03	2.45E+03
5	1	50.0	-7.02E+02	0.	-1.282E+03 -4.906E+03 -1.282E+03	2.3420E+03	-4.9055E+03	2.45E+03
5	202 JB	100.0	-1.53E+03	0.	-1.282E+03 2.350E+04 -1.282E+03	2.3500E+04	-2.6063E+04	1.30E+04
6	120 JA	0.0	-2.58E+02	0.	-1.215E+03 -1.476E+04 -1.215E+03	1.2326E+04	-1.4757E+04	7.38E+03 BRECT
6	1	50.0	-6.70E+02	0.	-1.215E+03 -2.933E+03 -1.215E+03	5.0256E+02	-2.9333E+03	1.47E+03
6	1	50.0	-6.70E+02	0.	-1.215E+03 -2.933E+03 -1.215E+03	5.0256E+02	-2.9333E+03	1.47E+03
6	218 JB	100.0	-1.08E+03	0.	-1.215E+03 1.939E+04 -1.215E+03	1.9390E+04	-2.1821E+04	1.09E+04

1
0 GRAVITY, VERTICAL UDL ON FLOORS, LATERAL LDL & POINT LOADS ON WALL 1
0 MAXIMUM STRESS SUMMARY FOR

— TRANSVERSE —				TORSIONAL (T*Q/J)	B E A M S T R E S S E S			FOR OUTPUT VECTOR		COMBINED SHEAR
SHEAR (V2/X3*A)		SHEAR (V3/X2*A)			- STRESSES AT USER LOCATIONS -			MAXIMUM (P/A+MC/I)	MINIMUM (P/A-MC/I)	
					A	B	C			
0	MAX. BEAM STRESSES=	3.14E+02	0.	0.	-3.693E+02	2.355E+04	-3.693E+02	2.3546E+04	-1.7942E+03	1.30E+04
	BEAM NOS. =	2.	1.	1.	2.	4.	2.	4.	1.	5.
0	MIN. BEAM STRESSES=	-1.53E+03	0.	0.	-1.282E+03	-1.476E+04	-1.282E+03	5.0256E+02	-2.6063E+04	8.97E+02
	BEAM NOS. =	5.	1.	1.	5.	6.	5.	6.	5.	1.

1
0 GRAVITY, VERTICAL UDL ON FLOORS, LATERAL LDL & POINT LOADS ON WALL 1

NODES	ELEMENT FORCES	QUADRILATERAL ELEMENT STRESSES	AND FORCES FOR OUTPUT VECTOR			ANGLE
			MAX	MIN	MAX SHEAR	
0 X2. 6	FX = 5.022E+01 FY = 6.375E+02 FXY = 1.018E+02 MX = 0. MY = 0. MXY = 0.	+ZFACE SX = 2.511E+02 SY = 3.187E+03 SXY = 5.089E+02 -ZFACE SX = 2.511E+02 SY = 3.187E+03 SXY = 5.089E+02	+ZFACE 3.273E+03 -ZFACE 3.273E+03	1.654E+02 1.654E+02	1.554E+03 1.554E+03	80.440 80.440
0 X2. 7	FX = -8.139E+00 FY = -1.183E+02 FXY = 1.312E+02 MX = 0. MY = 0. MXY = 0.	+ZFACE SX = -4.069E+01 SY = -5.916E+02 SXY = 6.561E+02 -ZFACE SX = -4.069E+01 SY = -5.916E+02 SXY = 6.561E+02	+ZFACE 3.954E+02 -ZFACE 3.954E+02	-1.028E+03 -1.028E+03	7.115E+02 1.272E+03	33.612 33.612
0 X2. 8	FX = -1.299E+02 FY = -8.138E+02 FXY = 1.094E+02 MX = 0. MY = 0. MXY = 0.	+ZFACE SX = -6.495E+02 SY = -4.069E+03 SXY = 5.468E+02 -ZFACE SX = -6.495E+02 SY = -4.069E+03 SXY = 5.468E+02	+ZFACE -5.642E+02 -ZFACE -5.642E+02	-4.154E+03 -4.154E+03	1.795E+03 3.903E+03	8.867 8.867
0 X2. 9	FX = -1.633E+02 FY = -1.764E+03 FXY = 8.385E+01 MX = 0. MY = 0. MXY = 0.	+ZFACE SX = -8.167E+02 SY = -8.819E+03 SXY = 4.193E+02 -ZFACE SX = -8.167E+02 SY = -8.819E+03 SXY = 4.193E+02	+ZFACE -7.948E+02 -ZFACE -7.948E+02	-8.841E+03 -8.841E+03	4.023E+03 8.471E+03	2.991 2.991
0 X2. 11	FX = -5.361E+00 FY = 5.277E+02 FXY = 7.876E+01 MX = 0. MY = 0. MXY = 0.	+ZFACE SX = -2.681E+01 SY = 2.639E+03 SXY = 3.938E+02 -ZFACE SX = -2.681E+01 SY = 2.639E+03 SXY = 3.938E+02	+ZFACE 2.695E+03 -ZFACE 2.695E+03	-8.377E+01 -8.377E+01	1.390E+03 2.738E+03	81.769 81.769
0 X2. 12	FX = 5.918E+00 FY = -1.530E+02 FXY = 1.791E+02 MX = 0. MY = 0. MXY = 0.	+ZFACE SX = 2.959E+01 SY = -7.649E+02 SXY = 8.957E+02 -ZFACE SX = 2.959E+01 SY = -7.649E+02 SXY = 8.957E+02	+ZFACE 6.121E+02 -ZFACE 6.121E+02	-1.347E+03 -1.347E+03	9.798E+02 1.736E+03	33.041 33.041
0 X2. 13	FX = -4.199E+01 FY = -7.263E+02 FXY = 2.275E+02 MX = 0. MY = 0. MXY = 0.	+ZFACE SX = -2.100E+02 SY = -3.632E+03 SXY = 1.140E+03 -ZFACE SX = -2.100E+02 SY = -3.632E+03 SXY = 1.140E+03	+ZFACE 1.349E+02 -ZFACE 1.349E+02	-3.976E+03 -3.976E+03	2.056E+03 4.046E+03	16.835 16.835
0 X2. 14	FX = -3.883E+01 FY = -1.683E+03 FXY = -5.961E+01 MX = 0. MY = 0. MXY = 0.	+ZFACE SX = -1.942E+02 SY = -8.416E+03 SXY = -2.981E+02 -ZFACE SX = -1.942E+02 SY = -8.416E+03 SXY = -2.981E+02	+ZFACE -1.834E+02 -ZFACE -1.834E+02	-8.427E+03 -8.427E+03	4.122E+03 8.337E+03	-2.073 -2.073
0 X2. 16	FX = 1.396E+00 FY = 4.347E+02	+ZFACE SX = 6.981E+00 SY = 2.174E+03	+ZFACE 2.251E+03 -ZFACE 2.251E+03	-7.013E+01 -7.013E+01	1.160E+03	79.497

0	11	17	FX = 8.319E+01	SXY = 4.159E+02	VM	2.287E+03	
	9	12	MY = 0.	-ZFACE SX = 6.981E+00			
	11	17	FX = -1.226E+01	SXY = 4.159E+02	-ZFACE	2.251E+03	79.497
	10	13	FX = -1.660E+02	SXY = -6.132E+01		1.160E+03	
	12	18	FX = 1.645E+02	SXY = 8.224E+02	VM	2.287E+03	
	10	13	MX = 0.	SXY = 8.224E+02	+ZFACE	4.621E+02	32.476
	12	18	MY = 0.	SX = -6.132E+01		1.634E+03	
	10	13	MX = 0.	SY = -8.300E+02	-ZFACE	4.621E+02	32.476
	12	18	MY = 0.	SXY = 8.224E+02		1.634E+03	
	10	13	MX = 0.	SX = 4.716E+02	+ZFACE	7.248E+02	12.654
	12	18	FX = 9.432E+01	SY = -4.298E+03		2.638E+03	
	11	14	FX = -8.597E+02	SXY = 1.128E+03	-ZFACE	4.954E+03	
	13	19	FX = 2.255E+02	SXY = 4.716E+02	+ZFACE	7.248E+02	12.654
	11	14	MX = 0.	SY = -4.298E+03		2.638E+03	
	13	19	MY = 0.	SXY = 1.128E+03	-ZFACE	7.248E+02	12.654
	11	14	MX = 0.	SX = 3.723E+02	+ZFACE	3.910E+02	-4.139
	13	19	FX = 7.447E+01	SY = -3.174E+03		1.792E+03	
	12	20	FX = -6.347E+02	SXY = -2.579E+02	VM	3.405E+03	
	15	21	FX = -5.159E+01	SXY = 3.723E+02	-ZFACE	3.910E+02	-4.139
	14	21	MX = 0.	SY = -3.174E+03		1.792E+03	
	16	22	FX = -3.645E+01	SXY = -2.579E+02	+ZFACE	1.921E+03	77.748
	13	17	FX = 3.644E+02	SY = 1.822E+03		1.101E+03	
	15	21	FX = 9.136E+01	SXY = 4.568E+02	VM	2.076E+03	
	13	17	MX = 0.	SX = -1.822E+02	-ZFACE	1.921E+03	77.748
	15	21	MY = 0.	SY = 1.822E+03		1.101E+03	
	16	22	MX = 0.	SXY = 4.568E+02	-ZFACE	1.921E+03	77.748
	14	21	FX = -1.364E+01	SXY = -6.819E+01	+ZFACE	4.646E+02	32.111
	17	23	FX = -1.776E+02	SY = -8.882E+02		1.701E+03	
	14	18	FX = 1.698E+02	SXY = 8.489E+02	-ZFACE	4.646E+02	32.111
	16	22	MX = 0.	SX = -6.819E+01		1.701E+03	
	18	24	FX = 8.159E+01	SXY = 8.489E+02	+ZFACE	4.458E+02	5.221
	15	19	FX = -8.185E+02	SY = -4.092E+03		2.288E+03	
	17	23	FX = 8.293E+01	SXY = 4.147E+02	VM	4.370E+03	
	15	19	MX = 0.	SX = 4.079E+02	-ZFACE	4.458E+02	5.221
	17	23	MY = 0.	SY = -4.092E+03		2.288E+03	
	18	24	MX = 0.	SXY = 4.147E+02	-ZFACE	4.458E+02	5.221
	16	22	FX = 3.545E+01	SX = 1.773E+02	+ZFACE	3.740E+02	13.593
	19	25	FX = -5.982E+02	SY = -2.991E+03		1.781E+03	
	16	20	FX = 1.627E+02	SXY = 8.136E+02	VM	3.390E+03	
	18	24	MX = 0.	SX = 1.773E+02	-ZFACE	3.740E+02	13.593
	20	26	MY = 0.	SY = -2.991E+03		1.781E+03	
	19	25	MX = 0.	SXY = 8.136E+02	-ZFACE	3.740E+02	13.593
	17	23	FX = -3.261E+01	SX = -1.631E+02	+ZFACE	1.829E+03	82.859
	20	26	FX = 3.595E+02	SY = 1.797E+03		1.011E+03	
	17	22	FX = 4.990E+01	SXY = 2.495E+02	-ZFACE	1.829E+03	82.859
	19	25	MX = 0.	SX = -1.631E+02		1.011E+03	
	21	27	MY = 0.	SY = 1.797E+03	-ZFACE	1.829E+03	82.859
	18	24	MX = 0.	SXY = 2.495E+02		1.933E+03	
	20	26	FX = -2.678E+00	SX = -1.339E+01	+ZFACE	4.391E+02	31.954
	22	28	FX = -1.448E+02	SY = -7.239E+02		8.077E+02	
	18	23	FX = 1.451E+02	SXY = 7.254E+02	VM	1.447E+03	
	20	26	MX = 0.	SX = -1.339E+01	-ZFACE	4.391E+02	31.954
	22	28	MY = 0.	SY = -7.239E+02		8.077E+02	
	19	24	MX = 0.	SXY = 7.254E+02	-ZFACE	4.391E+02	31.954
	21	27	FX = -4.135E+01	SX = -2.068E+02	+ZFACE	-1.145E+02	10.702
	23	29	FX = -5.395E+02	SY = -2.697E+03		1.338E+03	
	20	26	FX = 9.763E+01	SXY = 4.881E+02	VM	2.734E+03	
	22	28	MX = 0.	SX = -2.068E+02	-ZFACE	-1.145E+02	10.702
	24	30	MY = 0.	SY = -2.697E+03		1.338E+03	
	21	27	MX = 0.	SXY = 4.881E+02	-ZFACE	-1.145E+02	10.702
	23	29	FX = -8.325E+01	SX = -4.162E+02	+ZFACE	-2.573E+02	8.795
	25	31	FX = -1.379E+03	SY = -6.896E+03		3.399E+03	
	20	26	FX = 2.054E+02	SXY = 1.027E+03	VM	6.930E+03	
	22	28	MX = 0.	SX = -4.162E+02	-ZFACE	-2.573E+02	8.795
	24	30	MY = 0.	SY = -6.896E+03		3.399E+03	

[illegible]

82.-----X1	MX = 0.	SXY = 4.599E+02	WM 9.23E+02	
0 X2. 8E	FX = -3.637E+01	+ZFACE SX = -1.819E+02		
./QUAD/ 89	FY = -2.028E+02	SY = -1.014E+03	+ZFACE -1.662E+01 -1.179E+03	5.813E+02 22.14E
./ 67/	FX = 8.119E+01	SXY = 4.060E+02	WM 1.171E+03	
./ 84	MX = 0.	-ZFACE SX = -1.819E+02		
83.-----X1	MY = 0.	SY = -1.014E+03	-ZFACE -1.662E+01 -1.179E+03	5.813E+02 22.14E
0 X2. 89	MX = 0.	SXY = 4.060E+02	WM 1.171E+03	
./QUAD/ 90	FX = -5.568E+01	+ZFACE SX = -2.784E+02		
./ 66/	FY = -3.288E+02	SY = -1.644E+03	+ZFACE -1.843E+02 -1.738E+03	7.770E+02 14.24E
./ 85	FX = 7.414E+01	SXY = 3.707E+02	WM 1.654E+03	
84.-----X1	MX = 0.	-ZFACE SX = -2.784E+02		
0 X2. 91	MY = 0.	SY = -1.644E+03	-ZFACE -1.843E+02 -1.738E+03	7.770E+02 14.24E
./QUAD/ 92	MX = 0.	SXY = 3.707E+02	WM 1.654E+03	
./ 69/	FX = -2.416E+01	+ZFACE SX = -1.208E+02		
./ 87	FY = -4.806E+01	SY = -2.403E+02	+ZFACE 7.297E+01 -4.341E+02	2.535E+02 38.18E
86.-----X1	FX = 4.928E+01	SXY = 2.464E+02	WM 4.748E+02	
0 X2. 92	MX = 0.	-ZFACE SX = -1.208E+02		
./QUAD/ 93	MY = 0.	SY = -2.403E+02	-ZFACE 7.297E+01 -4.341E+02	2.535E+02 38.18E
./ 70/	FX = -7.523E+01	SXY = 2.464E+02	WM 4.748E+02	
./ 88	FY = -1.360E+02	+ZFACE SX = -3.761E+02		
87.-----X1	FX = 6.792E+01	SY = -6.802E+02	+ZFACE -1.561E+02 -9.003E+02	3.721E+02 32.94E
0 X2. 93	MX = 0.	SXY = 3.396E+02	WM 8.332E+02	
./QUAD/ 94	MY = 0.	-ZFACE SX = -3.761E+02		
./ 71/	FX = -5.573E+01	SY = -6.802E+02	-ZFACE -1.561E+02 -9.003E+02	3.721E+02 32.94E
./ 89	FX = -1.896E+02	SXY = 3.396E+02	WM 8.332E+02	
88.-----X1	FX = 8.223E+01	+ZFACE SX = -2.786E+02		
0 X2. 94	MY = 0.	SY = -9.481E+02	+ZFACE -8.321E+01 -1.144E+03	5.302E+02 25.42E
./QUAD/ 95	FX = -9.508E+00	SXY = 4.111E+02	WM 1.104E+03	
./ 72/	FY = -2.558E+02	-ZFACE SX = -2.786E+02		
./ 90	FX = -2.558E+02	SY = -9.481E+02	-ZFACE -8.321E+01 -1.144E+03	5.302E+02 25.42E
89.-----X1	FX = 3.308E+01	SXY = 4.111E+02	WM 1.104E+03	
0 X2. 96	MY = 0.	+ZFACE SX = -4.754E+01		
./QUAD/ 97	FX = -8.871E+01	SY = -1.279E+03	+ZFACE -2.571E+01 -1.301E+03	6.375E+02 7.51E
./ 73/	FY = -8.286E+01	SXY = 1.654E+02	WM 1.288E+03	
./ 92	FX = -8.286E+01	-ZFACE SX = -4.754E+01		
91.-----X1	FX = 9.586E+01	SY = -1.279E+03	-ZFACE -2.571E+01 -1.301E+03	6.375E+02 7.51E
0 X2. 97	MY = 0.	SXY = 1.654E+02	WM 1.288E+03	
./QUAD/ 98	FX = -8.871E+01	+ZFACE SX = -4.436E+02		
./ 74/	FY = -8.286E+01	SY = -4.143E+02	+ZFACE 5.063E+01 -9.085E+02	4.795E+02 45.87E
./ 93	FX = 9.586E+01	SXY = 4.793E+02	WM 9.348E+02	
92.-----X1	MX = 0.	-ZFACE SX = -4.436E+02		
0 X2. 98	MY = 0.	SY = -4.143E+02	-ZFACE 5.063E+01 -9.085E+02	4.795E+02 45.87E
./QUAD/ 99	FX = -1.043E+02	SXY = 4.793E+02	WM 9.348E+02	
./ 75/	FY = -1.277E+02	+ZFACE SX = -5.213E+02		
./ 94	FX = -1.043E+02	SY = -6.389E+02	+ZFACE -2.982E+02 -8.620E+02	2.819E+02 38.98E
93.-----X1	FX = 5.514E+01	SXY = 2.757E+02	WM 7.582E+02	
0 X2. 99	MY = 0.	-ZFACE SX = -5.213E+02		
./QUAD/ 100	FX = -3.456E+01	SY = -6.389E+02	-ZFACE -2.982E+02 -8.620E+02	2.819E+02 38.98E
./ 76/	FY = -2.023E+02	SXY = 2.757E+02	WM 7.582E+02	
./ 95	FX = -2.023E+02	+ZFACE SX = -1.728E+02		
94.-----X1	FX = 6.903E+01	SY = -1.011E+03	+ZFACE -4.897E+01 -1.135E+03	5.431E+02 19.73E
0 X2. 101	MY = 0.	SXY = 3.452E+02	WM 1.112E+03	
./QUAD/ 102	FX = -3.580E+01	-ZFACE SX = -1.728E+02		
./ 77/	FY = -1.288E+02	SY = -1.011E+03	-ZFACE -4.897E+01 -1.135E+03	5.431E+02 19.73E
./ 97	FX = -1.288E+02	SXY = 3.452E+02	WM 1.112E+03	
96.-----X1	FX = 5.282E+00	+ZFACE SX = -1.790E+02		
0 X2. 102	MY = 0.	SY = -6.442E+02	+ZFACE -1.775E+02 -6.457E+02	2.341E+02 3.23E
./QUAD/ 103	FX = -2.352E+02	SXY = 2.641E+01	WM 5.778E+02	
./ 78/	FY = -1.014E+02	-ZFACE SX = -1.790E+02		
./ 99	FX = -1.014E+02	SY = -6.442E+02	-ZFACE -1.775E+02 -6.457E+02	2.341E+02 3.23E
97.-----X1	FX = 1.435E+02	SXY = 2.641E+01	WM 5.778E+02	
0 X2. 103	MY = 0.	+ZFACE SX = -1.176E+03		
./QUAD/ 104	FX = -2.352E+02	SY = -5.069E+02	+ZFACE -4.990E+01 -1.633E+03	7.914E+02 57.49E
./ 79/	FY = -1.435E+02	SXY = 7.173E+02	WM 1.608E+03	
./ 100	FX = -1.435E+02	-ZFACE SX = -1.176E+03		
98.-----X1	FX = -7.287E+01	SY = -5.069E+02	-ZFACE -4.990E+01 -1.633E+03	7.914E+02 57.49E
0 X2. 104	MY = 0.	SXY = 7.173E+02	WM 1.608E+03	

97.	QUAD / 103	76	96	X1	FY = -1.135E+02	SY = -5.673E+02	+ZFACE	-3.597E+02	-5.720E+02	1.062E+02	-8.518
0	X2.	103			FX = -6.220E+00	SXY = -3.110E+01				5.008E+02	
					MX = 0.	-ZFACE	SX = -3.643E+02				
					MY = 0.		SY = -5.673E+02	-ZFACE	-3.597E+02	-5.720E+02	1.062E+02
					MOXY = 0.		SXY = -3.110E+01			5.008E+02	-8.518
					FX = -2.537E+01	+ZFACE	SX = -1.269E+02				
					FY = -2.024E+02		SY = -1.012E+03	+ZFACE	1.141E+02	-1.253E+03	6.836E+02
					FX = 1.042E+02		SXY = 5.209E+02			1.314E+03	24.824
					MX = 0.	-ZFACE	SX = -1.269E+02				
					MY = 0.		SY = -1.012E+03	-ZFACE	1.141E+02	-1.253E+03	6.836E+02
					MOXY = 0.		SXY = 5.209E+02			1.314E+03	24.824
					FX = -9.840E+01	+ZFACE	SX = -4.920E+02				
					FY = -1.174E+02		SY = -5.868E+02	+ZFACE	-4.534E+02	-6.253E+02	8.595E+01
					FX = -1.434E+01		SXY = -7.172E+01			5.596E+02	-28.275
					MX = 0.	-ZFACE	SX = -4.920E+02				
					MY = 0.		SY = -5.868E+02	-ZFACE	-4.534E+02	-6.253E+02	8.595E+01
					MOXY = 0.		SXY = -7.172E+01			5.596E+02	-28.275
					FX = -1.525E+02	+ZFACE	SX = -7.625E+02				
					FY = -1.143E+01		SY = -5.715E+01	+ZFACE	1.284E+02	-9.480E+02	5.382E+02
					FX = -8.131E+01		SXY = -4.066E+02			1.018E+03	-65.469
					MX = 0.	-ZFACE	SX = -7.625E+02				
					MY = 0.		SY = -5.715E+01	-ZFACE	1.284E+02	-9.480E+02	5.382E+02
					MOXY = 0.		SXY = -4.066E+02			1.018E+03	-65.469
					FX = -4.828E+01	+ZFACE	SX = -2.414E+02				
					FY = -8.828E+01		SY = -4.414E+02	+ZFACE	-1.598E+02	-5.230E+02	1.816E+02
					FX = 3.033E+01		SXY = 1.517E+02			4.643E+02	28.303
					MX = 0.	-ZFACE	SX = -2.414E+02				
					MY = 0.		SY = -4.414E+02	-ZFACE	-1.598E+02	-5.230E+02	1.816E+02
					MOXY = 0.		SXY = 1.517E+02			4.643E+02	28.303
					FX = -3.839E+01	+ZFACE	SX = -1.920E+02				
					FY = -1.230E+02		SY = -6.151E+02	+ZFACE	-2.749E+01	-7.796E+02	3.760E+02
					FX = 6.218E+01		SXY = 3.109E+02			7.662E+02	27.882
					MX = 0.	-ZFACE	SX = -1.920E+02				
					MY = 0.		SY = -6.151E+02	-ZFACE	-2.749E+01	-7.796E+02	3.760E+02
					MOXY = 0.		SXY = 3.109E+02			7.662E+02	27.882
					FX = -6.566E+01	+ZFACE	SX = -3.283E+02				
					FY = -8.535E+01		SY = -4.267E+02	+ZFACE	2.596E+01	-7.810E+02	4.035E+02
					FX = 8.009E+01		SXY = 4.005E+02			7.943E+02	41.496
					MX = 0.	-ZFACE	SX = -3.283E+02				
					MY = 0.		SY = -4.267E+02	-ZFACE	2.596E+01	-7.810E+02	4.035E+02
					MOXY = 0.		SXY = 4.005E+02			7.943E+02	41.496
					FX = -1.420E+01	+ZFACE	SX = -7.099E+01				
					FY = 1.711E+01		SY = 8.556E+01	+ZFACE	1.171E+02	-1.025E+02	1.098E+02
					FX = -1.540E+01		SXY = -7.699E+01			1.903E+02	-67.737
					MX = 0.	-ZFACE	SX = -7.099E+01				
					MY = 0.		SY = 8.556E+01	-ZFACE	1.171E+02	-1.025E+02	1.098E+02
					MOXY = 0.		SXY = -7.699E+01			1.903E+02	-67.737
					FX = -1.219E+02	+ZFACE	SX = -6.096E+02				
					FY = -1.474E+02		SY = -7.372E+02	+ZFACE	-4.610E+02	-8.857E+02	2.123E+02
					FX = 4.051E+01		SXY = 2.025E+02			7.673E+02	36.260
					MX = 0.	-ZFACE	SX = -6.096E+02				
					MY = 0.		SY = -7.372E+02	-ZFACE	-4.610E+02	-8.857E+02	2.123E+02
					MOXY = 0.		SXY = 2.025E+02			7.673E+02	36.260
					FX = -1.334E+02	+ZFACE	SX = -6.668E+02				
					FY = -1.321E+02		SY = -6.607E+02	+ZFACE	-5.053E+02	-8.222E+02	1.585E+02
					FX = 3.169E+01		SXY = 1.584E+02			7.183E+02	45.554
					MX = 0.	-ZFACE	SX = -6.668E+02				
					MY = 0.		SY = -6.607E+02	-ZFACE	-5.053E+02	-8.222E+02	1.585E+02
					MOXY = 0.		SXY = 1.584E+02			7.183E+02	45.554
					FX = -8.723E+00	+ZFACE	SX = -4.362E+01				
					FY = -2.207E+01		SY = -1.104E+02	+ZFACE	9.865E+01	-2.526E+02	1.756E+02
					FX = 3.449E+01		SXY = 1.724E+02			3.138E+02	39.524
					MX = 0.	-ZFACE	SX = -4.362E+01				
					MY = 0.		SY = -1.104E+02	-ZFACE	9.865E+01	-2.526E+02	1.756E+02
					MOXY = 0.		SXY = 1.724E+02			3.138E+02	39.524
					FX = -7.790E+01	+ZFACE	SX = -3.895E+02				
					FY = 3.519E+00		SY = 1.760E+01	+ZFACE	1.015E+02	-4.734E+02	2.874E+02
					FX = 4.058E+01		SXY = 2.029E+02			5.314E+02	67.545
					MX = 0.	-ZFACE	SX = -3.895E+02				

111.-----X1	MY = 0.	SY = 1.760E+01	-ZFACE	1.015E+02	-4.734E+02	2.874E+02	67.545
0 X2. 117	MAXY = 0.	SXY = 2.029E+02				WM 5.314E+02	
./QUAD / 118	FX = -2.916E+02	+ZFACE	SX = -1.458E+03				
./ 90/	FY = -1.591E+02		SY = -7.953E+02	+ZFACE	-4.362E+02	-1.817E+03	59.342
112.-----X1	FXY = 1.212E+02		SXY = 6.058E+02			WM 1.643E+03	
0 X2. 118	MX = 0.	-ZFACE	SX = -1.458E+03				
./QUAD / 113	MY = 0.	SY = -7.953E+02	-ZFACE	-4.362E+02	-1.817E+03	6.906E+02	59.342
./ 91/	MAXY = 0.	SXY = 6.058E+02				WM 1.643E+03	
113.-----X1	FX = -1.932E+02	+ZFACE	SX = -9.660E+02				
0 X2. 119	FY = -8.191E+01		SY = -4.095E+02	+ZFACE	-3.441E+02	-1.031E+03	-72.033
./QUAD / 114	FXY = -4.033E+01		SXY = -2.017E+02			WM 9.095E+02	
./ 91/	MX = 0.	-ZFACE	SX = -9.660E+02				
114.-----X1	MY = 0.	SY = -4.095E+02	-ZFACE	-3.441E+02	-1.031E+03	3.436E+02	-72.033
0 X2. 119	MAXY = 0.	SXY = -2.017E+02				WM 9.095E+02	
./QUAD / 120	FX = -1.470E+02	+ZFACE	SX = -7.350E+02				
./ 92/	FY = -7.910E+01		SY = -3.955E+02	+ZFACE	-2.604E+02	-8.702E+02	-61.918
115.-----X1	FXY = -5.065E+01		SXY = -2.533E+02			WM 7.736E+02	
0 X2. 121	MX = 0.	-ZFACE	SX = -7.350E+02				
./QUAD / 122	MY = 0.	SY = -3.955E+02	-ZFACE	-2.604E+02	-8.702E+02	3.049E+02	-61.918
./ 93/	MAXY = 0.	SXY = -2.533E+02				WM 7.736E+02	
116.-----X1	FX = -6.606E+02	+ZFACE	SX = -3.303E+03				
0 X2. 122	FY = -5.104E+01		SY = -2.552E+02	+ZFACE	1.071E+02	-3.665E+03	71.946
./QUAD / 123	FXY = 2.223E+02		SXY = 1.112E+03			WM 3.720E+03	
./ 93/	MX = 0.	-ZFACE	SX = -3.303E+03				
117.-----X1	MY = 0.	SY = -2.552E+02	-ZFACE	1.071E+02	-3.665E+03	1.886E+03	71.946
0 X2. 123	MAXY = 0.	SXY = 1.112E+03				WM 3.720E+03	
./QUAD / 124	FX = -2.880E+02	+ZFACE	SX = -1.440E+03				
./ 94/	FY = -9.838E+01		SY = -4.919E+02	+ZFACE	-4.820E+02	-1.450E+03	-84.191
118.-----X1	FXY = -1.950E+01		SXY = -9.748E+01			WM 1.279E+03	
0 X2. 124	MX = 0.	-ZFACE	SX = -1.440E+03				
./QUAD / 125	MY = 0.	SY = -4.919E+02	-ZFACE	-4.820E+02	-1.450E+03	4.841E+02	-84.191
./ 95/	MAXY = 0.	SXY = -9.748E+01				WM 1.279E+03	
119.-----X1	FX = -2.417E+02	+ZFACE	SX = -1.208E+03				
0 X2. 125	FY = -6.283E+01		SY = -3.142E+02	+ZFACE	-3.096E+02	-1.213E+03	85.940
./QUAD / 126	FXY = 1.276E+01		SXY = 6.379E+01			WM 1.091E+03	
./ 95/	MX = 0.	-ZFACE	SX = -1.208E+03				
120.-----X1	MY = 0.	SY = -3.142E+02	-ZFACE	-3.096E+02	-1.213E+03	4.516E+02	85.940
0 X2. 126	MAXY = 0.	SXY = 6.379E+01				WM 1.091E+03	
./QUAD / 127	FX = -3.776E+02	+ZFACE	SX = -1.888E+03				
./ 96/	FY = -8.498E+01		SY = -4.249E+02	+ZFACE	-3.618E+02	-1.951E+03	-78.500
121.-----X1	FXY = -6.210E+01		SXY = -3.105E+02			WM 1.798E+03	
0 X2. 127	MX = 0.	-ZFACE	SX = -1.888E+03				
./QUAD / 128	MY = 0.	SY = -4.249E+02	-ZFACE	-3.618E+02	-1.951E+03	7.946E+02	-78.500
./ 96/	MAXY = 0.	SXY = -3.105E+02				WM 1.798E+03	
122.-----X1	FX = 1.975E+01	+ZFACE	SX = 9.876E+01				
0 X2. 130	FY = 2.250E+02		SY = 1.125E+03	+ZFACE	1.327E+03	-1.035E+02	67.914
./QUAD / 131	FXY = 9.971E+01		SXY = 4.986E+02			WM 1.382E+03	
./ 97/	MX = 0.	-ZFACE	SX = 9.876E+01				
123.-----X1	MY = 0.	SY = 1.125E+03	-ZFACE	1.327E+03	-1.035E+02	7.155E+02	67.914
0 X2. 131	MAXY = 0.	SXY = 4.986E+02				WM 1.382E+03	
./QUAD / 132	FX = -8.026E+01	+ZFACE	SX = -4.013E+02				
./ 98/	FY = -6.739E+02		SY = -3.369E+03	+ZFACE	-2.118E+02	-3.559E+03	13.765
124.-----X1	FXY = 1.547E+02		SXY = 7.735E+02			WM 3.458E+03	
0 X2. 132	MX = 0.	-ZFACE	SX = -4.013E+02				
./QUAD / 133	MY = 0.	SY = -3.369E+03	-ZFACE	-2.118E+02	-3.559E+03	1.673E+03	13.765
./ 99/	MAXY = 0.	SXY = 7.735E+02				WM 3.458E+03	
125.-----X1	FX = -1.299E+02	+ZFACE	SX = -6.496E+02				
0 X2. 133	FY = -1.577E+03		SY = -7.883E+03	+ZFACE	-5.448E+02	-7.988E+03	6.815
./QUAD / 134	FXY = 1.754E+02		SXY = 8.770E+02			WM 7.730E+03	
./ 99/	MX = 0.	-ZFACE	SX = -6.496E+02				
126.-----X1	MY = 0.	SY = -7.883E+03	-ZFACE	-5.448E+02	-7.988E+03	3.722E+03	6.815
0 X2. 134	MAXY = 0.	SXY = 8.770E+02				WM 7.730E+03	
./QUAD / 135	FX = -6.006E+01	+ZFACE	SX = -3.003E+02				
./ 100/	FY = 1.044E+02		SY = 5.222E+02	+ZFACE	8.428E+02	-6.210E+02	62.093
127.-----X1	FXY = 1.211E+02		SXY = 6.054E+02			WM 1.273E+03	
0 X2. 135	MX = 0.	-ZFACE	SX = -3.003E+02				
./QUAD / 136	MY = 0.	SY = 5.222E+02	-ZFACE	8.428E+02	-6.210E+02	7.319E+02	62.093
./ 101/	MAXY = 0.	SXY = 6.054E+02				WM 1.273E+03	
128.-----X1	FX = -5.686E+01	+ZFACE	SX = -2.843E+02				

0 X2. 136	QUAD / 136	FX = -6.878E+02	FY = -6.878E+02	FX = -6.878E+02	FY = -6.878E+02	SX = -3.439E+03	SY = -3.439E+03	+ZFACE	2.994E+01	-3.753E+03	1.892E+03	16.751
	/ 101/	MX = 0.	MY = 0.	MX = 0.	MY = 0.	SXY = 1.044E+03	SXY = 1.044E+03	-ZFACE			3.768E+03	
0 X2. 136	QUAD / 132	FX = -1.218E+01	FY = -1.218E+01	FX = -1.218E+01	FY = -1.218E+01	SX = -2.843E+02	SY = -2.843E+02	+ZFACE	2.994E+01	-3.753E+03	1.892E+03	16.751
	/ 102/	MX = 0.	MY = 0.	MX = 0.	MY = 0.	SXY = 1.044E+03	SXY = 1.044E+03	-ZFACE			3.768E+03	
0 X2. 136	QUAD / 137	FX = -1.424E+03	FY = -1.424E+03	FX = -1.424E+03	FY = -1.424E+03	SX = -6.089E+01	SY = -6.089E+01	+ZFACE	-2.573E+01	-7.158E+03	3.566E+03	4.026
	/ 103/	MX = 0.	MY = 0.	MX = 0.	MY = 0.	SXY = 4.995E+02	SXY = 4.995E+02	-ZFACE			7.145E+03	
0 X2. 136	QUAD / 133	FX = -1.318E+02	FY = -1.318E+02	FX = -1.318E+02	FY = -1.318E+02	SX = -6.089E+01	SY = -6.089E+01	+ZFACE	-2.573E+01	-7.158E+03	3.566E+03	4.026
	/ 104/	MX = 0.	MY = 0.	MX = 0.	MY = 0.	SXY = 4.995E+02	SXY = 4.995E+02	-ZFACE			7.145E+03	
0 X2. 139	QUAD / 139	FX = -1.392E+02	FY = -1.392E+02	FX = -1.392E+02	FY = -1.392E+02	SX = -6.958E+02	SY = -6.958E+02	+ZFACE	-1.775E+02	-1.177E+03	4.998E+02	43.943
	/ 105/	MX = 0.	MY = 0.	MX = 0.	MY = 0.	SXY = 4.995E+02	SXY = 4.995E+02	-ZFACE			1.099E+03	
0 X2. 139	QUAD / 135	FX = -5.272E+01	FY = -5.272E+01	FX = -5.272E+01	FY = -5.272E+01	SX = -6.958E+02	SY = -6.958E+02	+ZFACE	-1.775E+02	-1.177E+03	4.998E+02	43.943
	/ 106/	MX = 0.	MY = 0.	MX = 0.	MY = 0.	SXY = 4.995E+02	SXY = 4.995E+02	-ZFACE			1.099E+03	
0 X2. 139	QUAD / 140	FX = -6.345E+02	FY = -6.345E+02	FX = -6.345E+02	FY = -6.345E+02	SX = -2.636E+02	SY = -2.636E+02	+ZFACE	-1.593E+01	-3.420E+03	1.702E+03	15.649
	/ 107/	MX = 0.	MY = 0.	MX = 0.	MY = 0.	SXY = 8.842E+02	SXY = 8.842E+02	-ZFACE			3.412E+03	
0 X2. 140	QUAD / 136	FX = -7.416E+00	FY = -7.416E+00	FX = -7.416E+00	FY = -7.416E+00	SX = -2.636E+02	SY = -2.636E+02	+ZFACE	-1.593E+01	-3.420E+03	1.702E+03	15.649
	/ 108/	MX = 0.	MY = 0.	MX = 0.	MY = 0.	SXY = 8.842E+02	SXY = 8.842E+02	-ZFACE			3.412E+03	
0 X2. 140	QUAD / 141	FX = -1.327E+03	FY = -1.327E+03	FX = -1.327E+03	FY = -1.327E+03	SX = -3.708E+01	SY = -3.708E+01	+ZFACE	-1.009E+01	-6.661E+03	3.325E+03	3.652
	/ 109/	MX = 0.	MY = 0.	MX = 0.	MY = 0.	SXY = 4.228E+02	SXY = 4.228E+02	-ZFACE			6.656E+03	
0 X2. 142	QUAD / 137	FX = -6.969E+01	FY = -6.969E+01	FX = -6.969E+01	FY = -6.969E+01	SX = -3.708E+01	SY = -3.708E+01	+ZFACE	-1.009E+01	-6.661E+03	3.325E+03	3.652
	/ 110/	MX = 0.	MY = 0.	MX = 0.	MY = 0.	SXY = 4.228E+02	SXY = 4.228E+02	-ZFACE			6.656E+03	
0 X2. 142	QUAD / 143	FX = -1.586E+02	FY = -1.586E+02	FX = -1.586E+02	FY = -1.586E+02	SX = -3.485E+02	SY = -3.485E+02	+ZFACE	-7.818E+01	-1.063E+03	4.925E+02	31.589
	/ 111/	MX = 0.	MY = 0.	MX = 0.	MY = 0.	SXY = 4.395E+02	SXY = 4.395E+02	-ZFACE			1.026E+03	
0 X2. 143	QUAD / 139	FX = -4.572E+01	FY = -4.572E+01	FX = -4.572E+01	FY = -4.572E+01	SX = -3.485E+02	SY = -3.485E+02	+ZFACE	-7.818E+01	-1.063E+03	4.925E+02	31.589
	/ 112/	MX = 0.	MY = 0.	MX = 0.	MY = 0.	SXY = 4.395E+02	SXY = 4.395E+02	-ZFACE			1.026E+03	
0 X2. 143	QUAD / 144	FX = -6.382E+02	FY = -6.382E+02	FX = -6.382E+02	FY = -6.382E+02	SX = -2.286E+02	SY = -2.286E+02	+ZFACE	-6.911E+01	-3.351E+03	1.641E+03	12.735
	/ 113/	MX = 0.	MY = 0.	MX = 0.	MY = 0.	SXY = 7.056E+02	SXY = 7.056E+02	-ZFACE			3.317E+03	
0 X2. 144	QUAD / 140	FX = -1.136E+01	FY = -1.136E+01	FX = -1.136E+01	FY = -1.136E+01	SX = -3.191E+03	SY = -3.191E+03	+ZFACE	-6.911E+01	-3.351E+03	1.641E+03	12.735
	/ 114/	MX = 0.	MY = 0.	MX = 0.	MY = 0.	SXY = 7.056E+02	SXY = 7.056E+02	-ZFACE			3.317E+03	
0 X2. 144	QUAD / 145	FX = -1.258E+03	FY = -1.258E+03	FX = -1.258E+03	FY = -1.258E+03	SX = -5.679E+01	SY = -5.679E+01	+ZFACE	-3.193E+01	-6.316E+03	3.142E+03	3.606
	/ 115/	MX = 0.	MY = 0.	MX = 0.	MY = 0.	SXY = 3.945E+02	SXY = 3.945E+02	-ZFACE			6.300E+03	
0 X2. 146	QUAD / 141	FX = 2.197E+01	FY = 2.197E+01	FX = 2.197E+01	FY = 2.197E+01	SX = -5.679E+01	SY = -5.679E+01	+ZFACE	-3.193E+01	-6.316E+03	3.142E+03	3.606
	/ 116/	MX = 0.	MY = 0.	MX = 0.	MY = 0.	SXY = 3.945E+02	SXY = 3.945E+02	-ZFACE			6.300E+03	
0 X2. 146	QUAD / 147	FX = 3.442E+01	FY = 3.442E+01	FX = 3.442E+01	FY = 3.442E+01	SX = 1.098E+02	SY = 1.098E+02	+ZFACE	7.427E+02	-4.607E+02	6.017E+02	46.483
	/ 117/	MX = 0.	MY = 0.	MX = 0.	MY = 0.	SXY = 6.009E+02	SXY = 6.009E+02	-ZFACE			1.052E+03	
0 X2. 147	QUAD / 143	FX = -1.660E+01	FY = -1.660E+01	FX = -1.660E+01	FY = -1.660E+01	SX = 1.098E+02	SY = 1.098E+02	+ZFACE	7.427E+02	-4.607E+02	6.017E+02	46.483
	/ 118/	MX = 0.	MY = 0.	MX = 0.	MY = 0.	SXY = 6.009E+02	SXY = 6.009E+02	-ZFACE			1.052E+03	
0 X2. 147	QUAD / 148	FX = -5.979E+02	FY = -5.979E+02	FX = -5.979E+02	FY = -5.979E+02	SX = -8.300E+01	SY = -8.300E+01	+ZFACE	5.815E+01	-3.130E+03	1.594E+03	12.145
	/ 119/	MX = 0.	MY = 0.	MX = 0.	MY = 0.	SXY = 6.559E+02	SXY = 6.559E+02	-ZFACE			3.160E+03	
0 X2. 148	QUAD / 144	FX = -5.063E+00	FY = -5.063E+00	FX = -5.063E+00	FY = -5.063E+00	SX = -8.300E+01	SY = -8.300E+01	+ZFACE	5.815E+01	-3.130E+03	1.594E+03	12.145
	/ 120/	MX = 0.	MY = 0.	MX = 0.	MY = 0.	SXY = 6.559E+02	SXY = 6.559E+02	-ZFACE			3.160E+03	
0 X2. 148	QUAD / 149	FX = -1.104E+03	FY = -1.104E+03	FX = -1.104E+03	FY = -1.104E+03	SX = -5.520E+03	SY = -5.520E+03	+ZFACE	-4.350E+00	-5.541E+03	2.768E+03	3.528
	/ 121/	MX = 0.	MY = 0.	MX = 0.	MY = 0.	SXY = 3.401E+02	SXY = 3.401E+02	-ZFACE			5.539E+03	
0 X2. 150	QUAD / 145	FX = -5.260E+01	FY = -5.260E+01	FX = -5.260E+01	FY = -5.260E+01	SX = -2.532E+01	SY = -2.532E+01	+ZFACE	-4.350E+00	-5.541E+03	2.768E+03	3.528
	/ 122/	MX = 0.	MY = 0.	MX = 0.	MY = 0.	SXY = 3.401E+02	SXY = 3.401E+02	-ZFACE			5.539E+03	
0 X2. 150	QUAD / 151	FX = -8.162E+01	FY = -8.162E+01	FX = -8.162E+01	FY = -8.162E+01	SX = -2.630E+02	SY = -2.630E+02	+ZFACE	2.870E+02	-9.581E+02	6.226E+02	41.654
	/ 123/	MX = 0.	MY = 0.	MX = 0.	MY = 0.	SXY = 6.183E+02	SXY = 6.183E+02	-ZFACE			1.129E+03	

176. ———— / 177	MY = 0.	SY = -2.738E+03	-ZFACE	3.734E+01	-2.767E+03	1.402E+03	5.840
0 X2. 182	MX = 0.	SXY = 2.839E+02				WM 2.786E+03	
178. ———— / 183	FX = -8.673E+01	+ZFACE SX = -4.336E+02					
0 X2. 182	FY = -2.138E+02	SY = -1.069E+03	+ZFACE	-1.012E+02	-1.401E+03	6.500E+02	30.377
179. ———— / 179	FXY = 1.134E+02	SXY = 5.672E+02				WM 1.354E+03	
0 X2. 183	MX = 0.	-ZFACE SX = -4.336E+02					
178. ———— / 184	MY = 0.	SY = -1.069E+03	-ZFACE	-1.012E+02	-1.401E+03	6.500E+02	30.377
0 X2. 183	MX = 0.	SXY = 5.672E+02				WM 1.354E+03	
179. ———— / 180	FX = 1.324E+00	+ZFACE SX = 6.618E+00					
0 X2. 184	FY = -3.099E+02	SY = -1.550E+03	+ZFACE	1.491E+02	-1.692E+03	9.206E+02	16.154
180. ———— / 181	FXY = 9.841E+01	SXY = 4.921E+02				WM 1.771E+03	
0 X2. 184	MX = 0.	-ZFACE SX = 6.618E+00					
179. ———— / 185	MY = 0.	SY = -1.550E+03	-ZFACE	1.491E+02	-1.692E+03	9.206E+02	16.154
0 X2. 185	MX = 0.	SXY = 4.921E+02				WM 1.771E+03	
180. ———— / 186	FX = -5.274E+00	+ZFACE SX = -2.637E+01					
0 X2. 186	FY = -4.794E+02	SY = -2.397E+03	+ZFACE	4.364E+00	-2.428E+03	1.216E+03	6.454
181. ———— / 187	FXY = 5.434E+01	SXY = 2.717E+02				WM 2.430E+03	
0 X2. 186	MX = 0.	-ZFACE SX = -2.637E+01					
180. ———— / 188	MY = 0.	SY = -2.397E+03	-ZFACE	4.364E+00	-2.428E+03	1.216E+03	6.454
0 X2. 187	MX = 0.	SXY = 2.717E+02				WM 2.430E+03	
181. ———— / 189	FX = -9.898E+01	+ZFACE SX = -4.949E+02					
0 X2. 187	FY = -1.443E+02	SY = -7.213E+02	+ZFACE	-4.766E+02	-7.397E+02	1.315E+02	-15.295
182. ———— / 190	FXY = -1.339E+01	SXY = -6.693E+01				WM 6.494E+02	
0 X2. 188	MX = 0.	-ZFACE SX = -4.949E+02					
183. ———— / 191	MY = 0.	SY = -7.213E+02	-ZFACE	-4.766E+02	-7.397E+02	1.315E+02	-15.295
0 X2. 188	MX = 0.	SXY = -6.693E+01				WM 6.494E+02	
184. ———— / 192	FX = 1.972E+00	+ZFACE SX = 9.858E+00					
0 X2. 189	FY = -3.205E+02	SY = -1.603E+03	+ZFACE	2.042E+02	-1.797E+03	1.001E+03	18.160
185. ———— / 193	FXY = 1.185E+02	SXY = 5.926E+02				WM 1.907E+03	
0 X2. 189	MX = 0.	-ZFACE SX = 9.858E+00					
186. ———— / 194	MY = 0.	SY = -1.603E+03	-ZFACE	2.042E+02	-1.797E+03	1.001E+03	18.160
0 X2. 190	MX = 0.	SXY = 5.926E+02				WM 1.907E+03	
187. ———— / 195	FX = -9.317E+00	+ZFACE SX = -4.658E+01					
0 X2. 190	FY = -4.321E+02	SY = -2.161E+03	+ZFACE	-2.080E+01	-2.187E+03	1.083E+03	6.264
188. ———— / 196	FXY = 4.698E+01	SXY = 2.349E+02				WM 2.176E+03	
0 X2. 191	MX = 0.	-ZFACE SX = -4.658E+01					
189. ———— / 197	MY = 0.	SY = -2.161E+03	-ZFACE	-2.080E+01	-2.187E+03	1.083E+03	6.264
0 X2. 191	MX = 0.	SXY = 2.349E+02				WM 2.176E+03	
190. ———— / 198	FX = 2.138E+01	+ZFACE SX = 1.069E+02					
0 X2. 192	FY = -1.496E+02	SY = -7.480E+02	+ZFACE	1.225E+02	-7.636E+02	4.430E+02	-7.621
191. ———— / 199	FXY = -2.329E+01	SXY = -1.165E+02				WM 8.316E+02	
0 X2. 192	MX = 0.	-ZFACE SX = 1.069E+02					
192. ———— / 200	MY = 0.	SY = -7.480E+02	-ZFACE	1.225E+02	-7.636E+02	4.430E+02	-7.621
0 X2. 193	MX = 0.	SXY = -1.165E+02				WM 8.316E+02	
193. ———— / 201	FX = -2.729E+01	+ZFACE SX = -1.365E+02					
0 X2. 193	FY = -3.325E+02	SY = -1.663E+03	+ZFACE	7.260E+01	-1.872E+03	9.722E+02	19.142
194. ———— / 202	FXY = 1.205E+02	SXY = 6.023E+02				WM 1.909E+03	
0 X2. 194	MX = 0.	-ZFACE SX = -1.365E+02					
195. ———— / 203	MY = 0.	SY = -1.663E+03	-ZFACE	7.260E+01	-1.872E+03	9.722E+02	19.142
0 X2. 194	MX = 0.	SXY = 6.023E+02				WM 1.909E+03	
196. ———— / 204	FX = -8.776E+00	+ZFACE SX = -4.388E+01					
0 X2. 195	FY = -4.133E+02	SY = -2.067E+03	+ZFACE	-1.999E+01	-2.090E+03	1.035E+03	6.167
197. ———— / 205	FXY = 4.422E+01	SXY = 2.211E+02				WM 2.081E+03	
0 X2. 195	MX = 0.	-ZFACE SX = -4.388E+01					
198. ———— / 206	MY = 0.	SY = -2.067E+03	-ZFACE	-1.999E+01	-2.090E+03	1.035E+03	6.167
0 X2. 196	MX = 0.	SXY = 2.211E+02				WM 2.081E+03	
199. ———— / 207	FX = 4.650E+01	+ZFACE SX = 2.325E+02					
0 X2. 196	FY = -1.408E+02	SY = -7.041E+02	+ZFACE	3.832E+02	-8.548E+02	6.190E+02	20.419
200. ———— / 208	FXY = 8.096E+01	SXY = 4.048E+02				WM 1.098E+03	
0 X2. 197	MX = 0.	-ZFACE SX = 2.325E+02					
201. ———— / 209	MY = 0.	SY = -7.041E+02	-ZFACE	3.832E+02	-8.548E+02	6.190E+02	20.419
0 X2. 197	MX = 0.	SXY = 4.048E+02				WM 1.098E+03	
202. ———— / 210	FX = -2.964E+01	+ZFACE SX = -1.482E+02					
0 X2. 198	FY = -2.461E+02	SY = -1.231E+03	+ZFACE	2.243E+01	-1.401E+03	7.115E+02	20.253
203. ———— / 211	FXY = 9.248E+01	SXY = 4.624E+02				WM 1.413E+03	
0 X2. 198	MX = 0.	-ZFACE SX = -1.482E+02					
204. ———— / 212	MY = 0.	SY = -1.231E+03	-ZFACE	2.243E+01	-1.401E+03	7.115E+02	20.253
0 X2. 199	MX = 0.	SXY = 4.624E+02				WM 1.413E+03	
205. ———— / 213	FX = 4.508E-01	+ZFACE SX = 2.254E+00					

192.-----X1	QUAD / 197	FX = -1.017E+02	FY = -3.046E+02	FX = -1.017E+02	FY = -3.046E+02	SX = -5.085E+02	SY = -1.523E+03	+ZFACE	3.468E+01	-1.555E+03	7.951E+02	8.210
0 X2. 198	147/193	FX = -1.017E+02	FY = -3.046E+02	FX = -1.017E+02	FY = -3.046E+02	SX = -5.085E+02	SY = -1.523E+03	-ZFACE	3.468E+01	-1.555E+03	7.951E+02	8.210
194.-----X1	QUAD / 199	FX = -7.741E+00	FY = -1.973E+02	FX = -7.741E+00	FY = -1.973E+02	SX = -3.871E+01	SY = -9.867E+02	+ZFACE	8.114E+01	-1.107E+03	5.938E+02	18.521
0 X2. 199	148/195	FX = -7.741E+00	FY = -1.973E+02	FX = -7.741E+00	FY = -1.973E+02	SX = -3.871E+01	SY = -9.867E+02	-ZFACE	8.114E+01	-1.107E+03	5.938E+02	18.521
195.-----X1	QUAD / 200	FX = -8.233E+00	FY = -2.529E+02	FX = -8.233E+00	FY = -2.529E+02	SX = -4.116E+01	SY = -1.264E+03	+ZFACE	-9.492E+00	-1.296E+03	6.433E+02	9.027
0 X2. 200	149/196	FX = -8.233E+00	FY = -2.529E+02	FX = -8.233E+00	FY = -2.529E+02	SX = -4.116E+01	SY = -1.264E+03	-ZFACE	-9.492E+00	-1.296E+03	6.433E+02	9.027
196.-----X1	QUAD / 201	FX = -1.231E+02	FY = -1.199E+02	FX = -1.231E+02	FY = -1.199E+02	SX = -6.155E+02	SY = -5.997E+02	+ZFACE	-4.677E+02	-7.475E+02	1.399E+02	-46.616
0 X2. 202	150/197	FX = -1.231E+02	FY = -1.199E+02	FX = -1.231E+02	FY = -1.199E+02	SX = -6.155E+02	SY = -5.997E+02	-ZFACE	-4.677E+02	-7.475E+02	1.399E+02	-46.616
198.-----X1	QUAD / 203	FX = -9.084E+00	FY = -2.114E+02	FX = -9.084E+00	FY = -2.114E+02	SX = -4.542E+01	SY = -1.057E+03	+ZFACE	1.239E+02	-1.226E+03	6.751E+02	20.738
0 X2. 203	151/199	FX = -9.084E+00	FY = -2.114E+02	FX = -9.084E+00	FY = -2.114E+02	SX = -4.542E+01	SY = -1.057E+03	-ZFACE	1.239E+02	-1.226E+03	6.751E+02	20.738
199.-----X1	QUAD / 204	FX = -1.353E+01	FY = -2.160E+02	FX = -1.353E+01	FY = -2.160E+02	SX = -6.765E+01	SY = -1.080E+03	+ZFACE	-4.652E+01	-1.101E+03	5.274E+02	8.136
0 X2. 204	152/200	FX = -1.353E+01	FY = -2.160E+02	FX = -1.353E+01	FY = -2.160E+02	SX = -6.765E+01	SY = -1.080E+03	-ZFACE	-4.652E+01	-1.101E+03	5.274E+02	8.136
200.-----X1	QUAD / 205	FX = -1.425E+00	FY = -1.233E+02	FX = -1.425E+00	FY = -1.233E+02	SX = -7.125E+00	SY = -6.164E+02	+ZFACE	5.198E+01	-6.755E+02	3.637E+02	-16.562
0 X2. 206	153/201	FX = -1.425E+00	FY = -1.233E+02	FX = -1.425E+00	FY = -1.233E+02	SX = -7.125E+00	SY = -6.164E+02	-ZFACE	5.198E+01	-6.755E+02	3.637E+02	-16.562
202.-----X1	QUAD / 207	FX = -4.050E+01	FY = -2.227E+02	FX = -4.050E+01	FY = -2.227E+02	SX = -1.113E+03	SY = -1.113E+03	+ZFACE	-1.739E+01	-1.298E+03	6.406E+02	22.340
0 X2. 207	154/203	FX = -4.050E+01	FY = -2.227E+02	FX = -4.050E+01	FY = -2.227E+02	SX = -1.113E+03	SY = -1.113E+03	-ZFACE	-1.739E+01	-1.298E+03	6.406E+02	22.340
203.-----X1	QUAD / 208	FX = -1.281E+01	FY = -2.015E+02	FX = -1.281E+01	FY = -2.015E+02	SX = 1.281E+02	SY = 1.281E+02	+ZFACE	-4.698E+01	-1.025E+03	4.889E+02	7.597
0 X2. 208	155/204	FX = -1.281E+01	FY = -2.015E+02	FX = -1.281E+01	FY = -2.015E+02	SX = 1.281E+02	SY = 1.281E+02	-ZFACE	-4.698E+01	-1.025E+03	4.889E+02	7.597
204.-----X1	QUAD / 209	FX = 2.882E+01	FY = -1.241E+02	FX = 2.882E+01	FY = -1.241E+02	SX = 1.441E+02	SY = -6.203E+02	+ZFACE	2.561E+02	-7.323E+02	4.942E+02	19.674
0 X2. 210	156/205	FX = 2.882E+01	FY = -1.241E+02	FX = 2.882E+01	FY = -1.241E+02	SX = 1.441E+02	SY = -6.203E+02	-ZFACE	2.561E+02	-7.323E+02	4.942E+02	19.674
206.-----X1	QUAD / 211	FX = -4.712E+01	FY = -1.283E+02	FX = -4.712E+01	FY = -1.283E+02	SX = -2.356E+02	SY = -6.415E+02	+ZFACE	-7.857E+01	-7.985E+02	3.600E+02	27.842
0 X2. 211	157/207	FX = -4.712E+01	FY = -1.283E+02	FX = -4.712E+01	FY = -1.283E+02	SX = -2.356E+02	SY = -6.415E+02	-ZFACE	-7.857E+01	-7.985E+02	3.600E+02	27.842
206.-----X1	QUAD / 212	FX = 0.	FY = 0.	FX = 0.	FY = 0.	SX = 0.	SY = 0.	+ZFACE	0.	0.	0.	0.
0 X2. 212	158/	FX = 0.	FY = 0.	FX = 0.	FY = 0.	SX = 0.	SY = 0.	-ZFACE	0.	0.	0.	0.

207. / 208	MY = 0.	SY = -6.415E+02	-ZFACE	-7.857E+01	-7.985E+02	3.600E+02	27.842
0 X2. 212	MX = 0.	SXY = 2.973E+02				7.622E+02	
./ QUAD / 213	FX = -6.313E+00	+ZFACE	SX = -3.196E+01				
./ 159/	FY = -1.092E+02		SY = -5.458E+02	+ZFACE	-1.123E+01	-5.662E+02	2.775E+02
208. / 209	FX = 2.085E+01		SXY = 1.043E+02			5.606E+02	11.036
0 X2. 214	MX = 0.	-ZFACE	SX = -3.156E+01				
./ QUAD / 215	MY = 0.	SY = -5.458E+02	-ZFACE	-1.123E+01	-5.662E+02	2.775E+02	11.036
./ 160/	MX = 0.	SXY = 1.043E+02				5.606E+02	
210. / 211	FX = -1.199E+02	+ZFACE	SX = -5.995E+02				
0 X2. 215	FY = -1.889E+02		SY = -9.444E+02	+ZFACE	-1.915E+02	-1.352E+03	5.804E+02
./ QUAD / 216	FX = 1.108E+02		SXY = 5.542E+02			1.268E+03	36.359
./ 161/	MX = 0.	-ZFACE	SX = -5.995E+02				
211. / 212	MY = 0.	SY = -9.444E+02	-ZFACE	-1.915E+02	-1.352E+03	5.804E+02	36.359
0 X2. 216	MX = 0.	SXY = 5.542E+02				1.268E+03	
./ QUAD / 217	FX = -2.960E+01	+ZFACE	SX = -1.480E+02				
./ 162/	FY = -6.469E+01		SY = -3.234E+02	+ZFACE	-8.752E+01	-3.839E+02	1.482E+02
212. / 213	FX = 2.389E+01		SXY = 1.195E+02			3.485E+02	26.855
0 X2. 218	MX = 0.	-ZFACE	SX = -1.480E+02				
./ QUAD / 219	MY = 0.	SY = -3.234E+02	-ZFACE	-8.752E+01	-3.839E+02	1.482E+02	26.855
./ 163/	MX = 0.	SXY = 1.195E+02				3.485E+02	
214. / 215	FX = -7.327E+00	+ZFACE	SX = -3.664E+01				
0 X2. 219	FY = -9.029E+01		SY = -4.515E+02	+ZFACE	-3.258E+01	-4.555E+02	2.115E+02
./ QUAD / 220	FX = 8.245E+00		SXY = 4.122E+01			4.401E+02	5.620
./ 164/	MX = 0.	-ZFACE	SX = -3.664E+01				
215. / 216	MY = 0.	SY = -4.515E+02	-ZFACE	-3.258E+01	-4.555E+02	2.115E+02	5.620
0 X2. 220	MX = 0.	SXY = 4.122E+01				4.401E+02	
./ QUAD / 221	FX = -1.287E+02	+ZFACE	SX = -6.437E+02				
./ 165/	FY = -4.732E+01		SY = -2.366E+02	+ZFACE	-2.273E+02	-6.530E+02	2.128E+02
216. / 217	FX = -1.244E+01		SXY = -6.222E+01			5.741E+02	-81.501
0 X2. 221	MX = 0.	-ZFACE	SX = -6.437E+02				
./ QUAD / 222	MY = 0.	SY = -2.366E+02	-ZFACE	-2.273E+02	-6.530E+02	2.128E+02	-81.501
./ 166/	MX = 0.	SXY = -6.222E+01				5.741E+02	
218. / 219	FX = -1.780E+01	+ZFACE	SX = -8.901E+01				
0 X2. 222	FY = -7.032E+01		SY = -3.516E+02	+ZFACE	-7.812E+01	-3.625E+02	1.422E+02
./ QUAD / 223	FX = 1.091E+01		SXY = 5.457E+01			3.304E+02	11.285
./ 167/	MX = 0.	-ZFACE	SX = -8.901E+01				
219. / 220	MY = 0.	SY = -3.516E+02	-ZFACE	-7.812E+01	-3.625E+02	1.422E+02	11.285
0 X2. 223	MX = 0.	SXY = 5.457E+01				3.304E+02	
./ QUAD / 224	FX = -3.394E+00	+ZFACE	SX = -1.697E+01				
./ 168/	FY = -7.604E+01		SY = -3.802E+02	+ZFACE	-1.642E+01	-3.808E+02	1.822E+02
220. / 221	FX = 2.828E+00		SXY = 1.414E+01			3.728E+02	2.226
0 X2. 224	MX = 0.	-ZFACE	SX = -1.697E+01				
./ QUAD / 225	MY = 0.	SY = -3.802E+02	-ZFACE	-1.642E+01	-3.808E+02	1.822E+02	2.226
./ 169/	MX = 0.	SXY = 1.414E+01				3.728E+02	
221. / 222	FX = 2.583E+01	+ZFACE	SX = 1.292E+02				
0 X2. 225	FY = -3.849E+01		SY = -1.925E+02	+ZFACE	1.716E+02	-2.349E+02	2.033E+02
./ QUAD / 226	FX = -2.486E+01		SXY = -1.243E+02			3.535E+02	-18.850
./ 170/	MX = 0.	-ZFACE	SX = 1.292E+02				
222. / 223	MY = 0.	SY = -1.925E+02	-ZFACE	1.716E+02	-2.349E+02	2.033E+02	-18.850
0 X2. 226	MX = 0.	SXY = -1.243E+02				3.535E+02	
./ QUAD / 227	FX = -1.922E+01	+ZFACE	SX = -9.612E+01				
./ 171/	FY = -7.797E+01		SY = -3.899E+02	+ZFACE	-9.462E+01	-3.913E+02	1.484E+02
223. / 224	FX = 4.208E+00		SXY = 2.104E+01			3.537E+02	4.076
0 X2. 227	MX = 0.	-ZFACE	SX = -9.612E+01				
./ QUAD / 228	MY = 0.	SY = -3.899E+02	-ZFACE	-9.462E+01	-3.913E+02	1.484E+02	4.076
./ 172/	MX = 0.	SXY = 2.104E+01				3.537E+02	
224. / 225	FX = -2.853E+00	+ZFACE	SX = -1.426E+01				
0 X2. 228	FY = -7.001E+01		SY = -3.500E+02	+ZFACE	-1.422E+01	-3.501E+02	1.679E+02
./ QUAD / 229	FX = 7.235E-01		SXY = 3.618E+00			3.432E+02	.617
./ 173/	MX = 0.	-ZFACE	SX = -1.426E+01				
225. / 226	MY = 0.	SY = -3.500E+02	-ZFACE	-1.422E+01	-3.501E+02	1.679E+02	.617
0 X2. 229	MX = 0.	SXY = 3.618E+00				3.432E+02	

1
0 GRAVITY, VERTICAL LDL ON FLOORS, LATERAL LDL & POINT LOADS ON WALL 1
0 QUADRILATERAL STRESSES AND FORCES FOR OUTPUT VECTOR 1
0
0
0 MAXIMUM SURFACE STRESS SUMMARY
0 MAXIMUM STRESS = 3.273111E+03 QUAD. NO. = 1.

0	MINIMUM STRESS	=	-8.840774E+03	QUAD. NO.	=	4.
0	MAXIMUM SHEAR STRESS	=	4.121900E+03	QUAD. NO.	=	8.
0	MAXIMUM VONMISES VALUE	=	8.471393E+03	QUAD. NO.	=	4.
1						

Appendix G

Dialogue with the computer during the execution of the postprocessor

STARDYNE POSTPROCESSOR FOR COUPLED SHEAR WALLS - METRIC VERSION

Enter name of file where STARDYNE output is stored:
?OSTRA

ERROR - File is not in directory or has incorrect access.
Enter correct file name:
?OSTAR

Enter name of file where postprocessor commands are stored:
?CPI

Reading data. Please wait ...

Do you want all data printed (Y/N)?
?Y

**** STOREY NO: 1 **** Storey drift = .975 mm

GEOMETRY:
Wall 1 width = 6.000 m Wall 2 width = 5.100 m
Beam length = 2.000 m Beam height = 1.500 m
Storey height = 4.000 m Thickness = .200 m

LOADS:
Lateral top LDL = .000 kN/m Bot. LDL = .000 kN/m
Conc. lat. load = 80.000 kN Vert. UDL = 70.000 kN/m

FORCES IN CONNECTING BEAM:

	At Wall 1	At midspan	At Wall 2
Axial (kN)	187.767	187.767	-187.767
Shear (kN)	-8.630	78.630	148.630
Moment (kN*m)	-55.313	11.683	-101.947

MAXIMUM STRESS SUMMARY (in MPa):

X	Y	XY	ELEMENT
-.8167E+00			4
	-.8819E+01		4
		.1140E+01	7

WALL 1:

ELEMENT	LOCATION		STRESSES		
	X (m)	Y (m)	X (MPa)	Y (MPa)	XY (MPa)
1	.750	.625	.2511E+00	.3187E+01	.5089E+00
2	2.250	.625	-.4069E-01	-.5916E+00	.6561E+00
3	3.750	.625	-.6495E+00	-.4069E+01	.5468E+00
4	5.250	.625	-.8167E+00	-.8819E+01	.4193E+00
5	.750	1.875	-.2681E-01	.2639E+01	.3938E+00
6	2.250	1.875	.2959E-01	-.7649E+00	.8957E+00
7	3.750	1.875	-.2100E+00	-.3632E+01	.1140E+01
8	5.250	1.875	-.1942E+00	-.8416E+01	-.2981E+00
9	.750	2.875	.6981E-02	.2174E+01	.4159E+00
10	2.250	2.875	-.6132E-01	-.8300E+00	.8224E+00
11	3.750	2.875	.4716E+00	-.4298E+01	.1128E+01
12	5.250	2.875	.3723E+00	-.3174E+01	-.2579E+00
13	.750	3.625	-.1822E+00	.1822E+01	.4568E+00
14	2.250	3.625	-.6819E-01	-.8882E+00	.8489E+00
15	3.750	3.625	.4079E+00	-.4092E+01	.4147E+00
16	5.250	3.625	.1773E+00	-.2991E+01	.8136E+00

WALL 2:

ELEMENT	LOCATION		STRESSES		
	X (m)	Y (m)	X (MPa)	Y (MPa)	XY (MPa)
97	.850	.625	.9876E-01	.1125E+01	.4986E+00
98	2.550	.625	-.4013E+00	-.3369E+01	.7735E+00
99	4.250	.625	-.6496E+00	-.7883E+01	.8770E+00
100	.850	1.875	-.3003E+00	.5222E+00	.6054E+00
101	2.550	1.875	-.2843E+00	-.3439E+01	.1044E+01
102	4.250	1.875	-.6089E-01	-.7122E+01	.4995E+00
103	.850	2.875	-.6589E+00	-.6958E+00	.4995E+00
104	2.550	2.875	-.2636E+00	-.3172E+01	.8842E+00
105	4.250	2.875	-.3708E-01	-.6634E+01	.4228E+00
106	.850	3.625	-.3485E+00	-.7930E+00	.4395E+00
107	2.550	3.625	-.2286E+00	-.3191E+01	.7056E+00
108	4.250	3.625	-.5679E-01	-.6291E+01	.3945E+00

**** STOREY NO: 2 ****

Storey drift = 2.298 mm

GEOMETRY:

Wall 1 width = 6.000 m Wall 2 width = 5.100 m
 Beam length = 2.000 m Beam height = .500 m
 Storey height = 3.000 m Thickness = .200 m

LOADS:

Lateral top LDL = .000 kN/m Bot. LDL = .000 kN/m
 Conc. lat. load = 150.000 kN Vert. UDL = 70.000 kN/m

FORCES IN CONNECTING BEAM:

	At Wall 1	At midspan	At Wall 2
Axial (kN)	36.933	36.933	-36.933
Shear (kN)	26.656	43.344	113.344
Moment (kN*m)	-20.359	12.015	-66.330

MAXIMUM STRESS SUMMARY (in MPa):

X	Y	XY	ELEMENT
-.5167E+00			29
	-.6896E+01		20
		.1027E+01	20

WALL 1:

ELEMENT	LOCATION		STRESSES		
	X (m)	Y (m)	X (MPa)	Y (MPa)	XY (MPa)
17	.750	4.625	-.1631E+00	.1797E+01	.2495E+00
18	2.250	4.625	-.1339E-01	-.7239E+00	.7254E+00
19	3.750	4.625	-.2068E+00	-.2697E+01	.4881E+00
20	5.250	4.625	-.4162E+00	-.6896E+01	.1027E+01
21	.750	5.875	-.8707E-01	.1344E+01	.3694E+00
22	2.250	5.875	-.2251E+00	-.7958E+00	.6848E+00
23	3.750	5.875	-.3867E+00	-.2868E+01	.9818E+00
24	5.250	5.875	-.1986E+00	-.6083E+01	.4543E+00
25	.750	6.625	-.1806E+00	.1034E+01	.4986E+00
26	2.250	6.625	-.4557E+00	-.7479E+00	.7100E+00
27	3.750	6.625	-.1010E+00	-.3574E+01	.9651E+00
28	5.250	6.625	.2918E+00	-.2771E+01	.2433E+00
29	.750	6.875	-.5167E+00	.9124E+00	.5926E+00
30	2.250	6.875	-.3804E+00	-.7574E+00	.6808E+00
31	3.750	6.875	-.5021E-01	-.3526E+01	.8423E+00
32	5.250	6.875	.1090E+00	-.2691E+01	.2995E+00

WALL 2:

ELEMENT	LOCATION		STRESSES		
	X (m)	Y (m)	X (MPa)	Y (MPa)	XY (MPa)
109	.850	4.625	.1098E+00	.1721E+00	.6009E+00
110	2.550	4.625	-.8300E-01	-.2989E+01	.6559E+00
111	4.250	4.625	-.2532E-01	-.5520E+01	.3401E+00
112	.850	5.875	-.2630E+00	-.4081E+00	.6183E+00
113	2.550	5.875	.4189E-01	-.2740E+01	.6556E+00
114	4.250	5.875	-.1163E-01	-.5100E+01	.3229E+00
115	.850	6.625	-.4068E+00	-.7472E+00	.1196E+00
116	2.550	6.625	.4379E-01	-.2619E+01	.7530E+00
117	4.250	6.625	-.1494E-01	-.4841E+01	.3010E+00
118	.850	6.875	.9554E-01	-.7804E+00	.8143E-01
119	2.550	6.875	-.8962E-01	-.2667E+01	.7517E+00
120	4.250	6.875	-.1229E-01	-.4730E+01	.2979E+00

***** STOREY NO: 3 *****

Storey drift = 3.991 mm

GEOMETRY:

Wall 1 width = 6.000 m	Wall 2 width = 5.100 m
Beam length = 2.000 m	Beam height = .500 m
Storey height = 3.000 m	Thickness = .200 m

LOADS:

Lateral top LDL = 2.000 kN/m	Bot. LDL = .000 kN/m
Conc. lat. load = 180.000 kN	Vert. UDL = 70.000 kN/m

FORCES IN CONNECTING BEAM:

	At Wall 1	At midspan	At Wall 2
Axial (kN)	53.474	53.474	-53.474
Shear (kN)	16.863	53.137	123.137
Moment (kN*m)	-30.154	12.017	-76.120

MAXIMUM STRESS SUMMARY (in MPa):

X	Y	XY	ELEMENT
-.6181E+00			45
	-.4703E+01		36
		.7826E+00	33

WALL 1:
=====

ELEMENT	LOCATION		STRESSES		
	X (m)	Y (m)	X (MPa)	Y (MPa)	XY (MPa)
33	.750	7.625	-.3821E+00	.1048E+01	.1484E+00
34	2.250	7.625	-.1833E+00	-.6967E+00	.7661E+00
35	3.750	7.625	-.1636E+00	-.2389E+01	.6733E+00
36	5.250	7.625	-.2235E+00	-.4703E+01	.5256E+00
37	.750	8.875	-.8882E-01	.6345E+00	.3770E+00
38	2.250	8.875	-.2513E+00	-.8818E+00	.6747E+00
39	3.750	8.875	-.2316E+00	-.2249E+01	.7826E+00
40	5.250	8.875	-.4108E-01	-.4127E+01	.2791E+00
41	.750	9.625	-.2339E+00	.3233E+00	.5236E+00
42	2.250	9.625	-.3869E+00	-.8292E+00	.6520E+00
43	3.750	9.625	-.8328E-01	-.2623E+01	.7353E+00
44	5.250	9.625	-.1911E+00	-.1928E+01	.9131E-01
45	.750	9.875	-.6181E+00	.2019E+00	.6432E+00
46	2.250	9.875	-.3198E+00	-.8125E+00	.5486E+00
47	3.750	9.875	-.3367E-01	-.2567E+01	.7078E+00
48	5.250	9.875	-.8175E-01	-.1859E+01	.1012E+00

WALL 2:
=====

ELEMENT	LOCATION		STRESSES		
	X (m)	Y (m)	X (MPa)	Y (MPa)	XY (MPa)
121	.850	7.625	.2504E+00	-.2880E+00	.5399E+00
122	2.550	7.625	-.8105E-01	-.2351E+01	.6292E+00
123	4.250	7.625	.1567E-01	-.4096E+01	.3191E+00
124	.850	8.875	-.3690E+00	-.8267E+00	.5845E+00
125	2.550	8.875	.2542E-01	-.2119E+01	.5893E+00
126	4.250	8.875	-.2041E-01	-.3701E+01	.3144E+00
127	.850	9.625	-.4391E+00	-.7610E+00	.1119E-01
128	2.550	9.625	.1889E-01	-.2114E+01	.6884E+00
129	4.250	9.625	-.3539E-01	-.3438E+01	.2836E+00
130	.850	9.875	.1282E+00	-.7929E+00	-.2752E-01
131	2.550	9.875	-.1202E+00	-.2171E+01	.6948E+00
132	4.250	9.875	-.3439E-01	-.3332E+01	.2731E+00

 ***** STOREY NO: 4 *****

Storey drift = 5.901 mm

GEOMETRY:

Wall 1 width = 6.000 m Wall 2 width = 5.100 m
 Beam length = 2.000 m Beam height = .500 m
 Storey height = 3.000 m Thickness = .200 m

LOADS:

Lateral top LDL = .000 kN/m Bot. LDL = .000 kN/m
 Conc. lat. load = 240.000 kN Vert. UDL = 70.000 kN/m

FORCES IN CONNECTING BEAM:

	At Wall 1	At midspan	At Wall 2
Axial (kN)	81.261	81.261	-81.261
Shear (kN)	11.766	56.234	126.234
Moment (kN*m)	-35.273	12.039	-81.195

MAXIMUM STRESS SUMMARY (in MPa):

X	Y	XY	ELEMENT
-.8394E+00			61
	-.3031E+01		52
		.6679E+00	61

WALL 1:

=====

ELEMENT	LOCATION		STRESSES		
	X (m)	Y (m)	X (MPa)	Y (MPa)	XY (MPa)
49	.750	10.625	-.4355E+00	.3751E+00	.8090E-01
50	2.250	10.625	-.1764E+00	-.6719E+00	.6426E+00
51	3.750	10.625	-.1417E+00	-.1667E+01	.5372E+00
52	5.250	10.625	-.2456E+00	-.3031E+01	.4310E+00
53	.750	11.875	-.1034E+00	.4357E-01	.3352E+00
54	2.250	11.875	-.3047E+00	-.8080E+00	.5450E+00
55	3.750	11.875	-.2565E+00	-.1542E+01	.5925E+00
56	5.250	11.875	-.2745E-01	-.2571E+01	.2190E+00
57	.750	12.625	-.3257E+00	-.2186E+00	.5187E+00
58	2.250	12.625	-.4540E+00	-.7545E+00	.5109E+00
59	3.750	12.625	-.1527E+00	-.1728E+01	.5403E+00
60	5.250	12.625	.2057E-01	-.1219E+01	.1981E-01
61	.750	12.875	-.8394E+00	-.3314E+00	.6879E+00
62	2.250	12.875	-.3611E+00	-.7114E+00	.3267E+00
63	3.750	12.875	-.1025E+00	-.1688E+01	.6007E+00
64	5.250	12.875	-.2919E+00	-.1156E+01	-.2115E-01

WALL 2:

=====

ELEMENT	LOCATION		STRESSES		
	X (m)	Y (m)	X (MPa)	Y (MPa)	XY (MPa)
133	.850	10.625	.2525E+00	-.5802E+00	.4845E+00
134	2.550	10.625	-.1168E+00	-.1786E+01	.5625E+00
135	4.250	10.625	.8305E-02	-.2738E+01	.2839E+00
136	.850	11.875	-.4336E+00	-.1069E+01	.5672E+00
137	2.550	11.875	.6618E-02	-.1550E+01	.4921E+00
138	4.250	11.875	-.2637E-01	-.2397E+01	.2717E+00
139	.850	12.625	-.4949E+00	-.7213E+00	-.6693E-01
140	2.550	12.625	.9858E-02	-.1603E+01	.5926E+00
141	4.250	12.625	-.4658E-01	-.2161E+01	.2349E+00
142	.850	12.875	.1069E+00	-.7480E+00	-.1165E+00
143	2.550	12.875	-.1365E+00	-.1663E+01	.6023E+00
144	4.250	12.875	-.4388E-01	-.2067E+01	.2211E+00

 ***** STOREY NO: 5 *****

Storey drift = 7.888 mm

GEOMETRY:

Wall 1 width = 6.000 m Wall 2 width = 5.100 m
 Beam length = 2.000 m Beam height = .500 m
 Storey height = 3.000 m Thickness = .200 m

LOADS:

Lateral top LDL = .000 kN/m Bot. LDL = .000 kN/m
 Conc. lat. load = 340.000 kN Vert. UDL = 70.000 kN/m

FORCES IN CONNECTING BEAM:

	At Wall 1	At midspan	At Wall 2
Axial (kN)	128.176	128.176	-128.176
Shear (kN)	10.317	59.684	129.683
Moment (kN·m)	-36.763	12.079	-82.604

MAXIMUM STRESS SUMMARY (in MPa):

X	Y	XY	ELEMENT
-1.1176E+01			77
	-1.1644E+01		68
		.7173E+00	77

WALL 1:

ELEMENT	LOCATION		STRESSES		
	X (m)	Y (m)	X (MPa)	Y (MPa)	XY (MPa)
65	.750	13.625	-.5724E+00	-.7042E-01	-.7409E-01
66	2.250	13.625	-.2258E+00	-.5366E+00	.4599E+00
67	3.750	13.625	-.1819E+00	-.1014E+01	.4060E+00
68	5.250	13.625	-.2784E+00	-.1644E+01	.3707E+00
69	.750	14.875	-.1208E+00	-.2403E+00	.2464E+00
70	2.250	14.875	-.3761E+00	-.6802E+00	.3396E+00
71	3.750	14.875	-.2786E+00	-.9481E+00	.4111E+00
72	5.250	14.875	-.4754E-01	-.1279E+01	.1654E+00
73	.750	15.625	-.4436E+00	-.4143E+00	.4793E+00
74	2.250	15.625	-.5213E+00	-.6389E+00	.2757E+00
75	3.750	15.625	-.1728E+00	-.1011E+01	.3452E+00
76	5.250	15.625	-.1790E+00	-.6442E+00	.2641E-01
77	.750	15.875	-.1176E+01	-.5069E+00	.7173E+00
78	2.250	15.875	-.3643E+00	-.5673E+00	-.3110E-01
79	3.750	15.875	-.1269E+00	-.1012E+01	.5209E+00
80	5.250	15.875	-.4920E+00	-.5868E+00	-.7172E-01

WALL 2:

ELEMENT	LOCATION		STRESSES		
	X (m)	Y (m)	X (MPa)	Y (MPa)	XY (MPa)
145	.850	13.625	.2325E+00	-.7041E+00	.4048E+00
146	2.550	13.625	-.1482E+00	-.1231E+01	.4624E+00
147	4.250	13.625	.2254E-02	-.1523E+01	.2248E+00
148	.850	14.875	-.5085E+00	-.1119E+01	.5348E+00

149	2.550	14.875	-.3871E-01	-.9867E+00	.3577E+00
150	4.250	14.875	-.4116E-01	-.1264E+01	.1994E+00
151	.850	15.625	-.6155E+00	-.5997E+00	-.1396E+00
152	2.550	15.625	-.4542E-01	-.1057E+01	.4471E+00
153	4.250	15.625	-.6765E-01	-.1080E+01	.1000E+00
154	.850	15.875	-.7125E-02	-.6164E+00	.0000E+00
155	2.550	15.875	-.2025E+00	-.1113E+01	.004E+00
156	4.250	15.875	-.6407E-01	-.1008E+01	.1281E+00

 ***** STOREY NO: 6 *****

Storey drift = 10.181 mm

GEOMETRY:

Wall 1 width = 6.000 m Wall 2 width = 5.100 m
 Beam length = 2.000 m Beam height = 1.000 m
 Storey height = 3.500 m Thickness = .200 m

LOADS:

Lateral top LDL = .000 kN/m Bot. LDL = .000 kN/m
 Conc. lat. load = 380.000 kN Vert. UDL = 70.000 kN/m

FORCES IN CONNECTING BEAM:

	At Wall 1	At midspan	At Wall 2
Axial (kN)	243.074	243.074	-243.074
Shear (kN)	-43.825	113.825	183.825
Moment (kN*m)	-90.278	11.453	-137.372

MAXIMUM STRESS SUMMARY (in MPa):

X	Y	XY	ELEMENT
-.3303E+01			93
	-.9444E+00		160
		.1112E+01	93

WALL 1: *****

ELEMENT	LOCATION		STRESSES		
	X (m)	Y (m)	X (MPa)	Y (MPa)	XY (MPa)
81	.750	16.625	-.7625E+00	-.5715E-01	-.4066E+00
82	2.250	16.625	-.2414E+00	-.4414E+00	.1517E+00
83	3.750	16.625	-.1920E+00	-.6151E+00	.3109E+00
84	5.250	16.625	-.3283E+00	-.4267E+00	.4005E+00
85	.750	17.875	-.7099E-01	.8556E-01	-.7699E-01
86	2.250	17.875	-.6096E+00	-.7372E+00	.2025E+00
87	3.750	17.875	-.6668E+00	-.6607E+00	.1584E+00
88	5.250	17.875	-.4362E-01	-.1104E+00	.1724E+00
89	.750	18.750	-.3895E+00	.1760E-01	.2029E+00
90	2.250	18.750	-.1458E+01	-.7953E+00	.6058E+00
91	3.750	18.750	-.9660E+00	-.4095E+00	-.2017E+00
92	5.250	18.750	-.7350E+00	-.3955E+00	-.2533E+00
93	.750	19.250	-.3303E+01	-.2552E+00	.1112E+01
94	2.250	19.250	-.1440E+01	-.4919E+00	-.9748E-01
95	3.750	19.250	-.1208E+01	-.3142E+00	.6379E-01
96	5.250	19.250	-.1888E+01	-.4249E+00	-.3105E+00

WALL 2:

ELEMENT	LOCATION		STRESSES		
	X (m)	Y (m)	X (MPa)	Y (MPa)	XY (MPa)
157	.850	16.625	.1441E+00	-.6203E+00	.3134E+00
158	2.550	16.625	-.2356E+00	-.6415E+00	.2973E+00
159	4.250	16.625	-.3156E-01	-.5458E+00	.1043E+00
160	.850	17.875	-.5995E+00	-.9444E+00	.5542E+00
161	2.550	17.875	-.1480E+00	-.3234E+00	.1195E+00
162	4.250	17.875	-.3664E-01	-.4515E+00	.4122E-01
163	.850	18.750	-.6437E+00	-.2366E+00	-.6222E-01
164	2.550	18.750	-.8901E-01	-.3516E+00	.5457E-01
165	4.250	18.750	-.1697E-01	-.3802E+00	.1414E-01
166	.850	19.250	.1292E+00	-.1925E+00	-.1243E+00
167	2.550	19.250	-.9612E-01	-.3899E+00	.2104E-01
168	4.250	19.250	-.1426E-01	-.3500E+00	.3618E-02

Do you wish to display data at different increments (Y/N)?
?Y

Do you want all data printed (Y/N)?
?N

After the following prompts, enter the starting value, end value, and increment for the information you want processed.

Structure has 6 storeys.
Enter start, end, and increment for storeys processed:
?2,6,2

Each storey has 4 elements in vertical direction.
Enter start, end, and increment for elements processed:
?1,4,4

Wall 1 has 4 elements in horizontal direction.
Enter start, end, and increment for elements processed:
?1,4,1

Wall 2 has 3 elements in horizontal direction.
Enter start, end, and increment for elements processed:
?1,3,3

**** STOREY NO: 2 **** Storey drift = 2.298 mm

GEOMETRY:
Wall 1 width = 6.000 m Wall 2 width = 5.100 m
Beam length = 2.000 m Beam height = .500 m
Storey height = 3.000 m Thickness = .200 m

LOADS:
Lateral top LDL = .000 kN/m Bot. LDL = .000 kN/m
Conc. lat. load = 150.000 kN Vert. UDL = 70.000 kN/m

FORCES IN CONNECTING BEAM:

	At Wall 1	At midspan	At Wall 2
Axial (kN)	36,933	36,933	-36,933
Shear (kN)	26,656	43,344	113,344
Moment (kN*m)	-20,359	12,015	-66,330

MAXIMUM STRESS SUMMARY (in MPa):

X	Y	XY	ELEMENT
-.5167E+00			29
	-.6896E+01		20
		.1027E+01	20

WALL 1:

ELEMENT	LOCATION		STRESSES		
	X (m)	Y (m)	X (MPa)	Y (MPa)	XY (MPa)
17	.750	4.625	-.1631E+00	.1797E+01	.2495E+00
18	2.250	4.625	-.1339E-01	-.7239E+00	.7254E+00
19	3.750	4.625	-.2068E+00	-.2697E+01	.4881E+00
20	5.250	4.625	-.4162E+00	-.6896E+01	.1027E+01

WALL 2:

ELEMENT	LOCATION		STRESSES		
	X (m)	Y (m)	X (MPa)	Y (MPa)	XY (MPa)
109	.850	4.625	.1098E+00	.1721E+00	.6009E+00

**** STOREY NO: 4 ****

Storey drift = 5.901 mm

GEOMETRY:

Wall 1 width = 6.000 m Wall 2 width = 5.100 m
 Beam length = 2.000 m Beam height = .500 m
 Storey height = 3.000 m Thickness = .200 m

LOADS:

Lateral top LDL = .000 kN/m Bot. LDL = .000 kN/m
 Conc. lat. load = 240,000 kN Vert. UDL = 70,000 kN/m

FORCES IN CONNECTING BEAM:

	At Wall 1	At midspan	At Wall 2
Axial (kN)	81,261	81,261	-81,261
Shear (kN)	11,766	58,234	128,234
Moment (kN*m)	-35,273	12,039	-81,195

MAXIMUM STRESS SUMMARY (in MPa):

X	Y	XY	ELEMENT
-.8394E+00			61
	-.3031E+01		52
		.6879E+00	61

WALL 1:

ELEMENT	LOCATION		STRESSES		
	X (m)	Y (m)	X (MPa)	Y (MPa)	XY (MPa)
49	.750	10.625	-.4355E+00	.3751E+00	.8090E-01
50	2.250	10.625	-.1764E+00	-.6719E+00	.6426E+00
51	3.750	10.625	-.1417E+00	-.1667E+01	.5372E+00
52	5.250	10.625	-.2456E+00	-.3031E+01	.4310E+00

WALL 2:

ELEMENT	LOCATION		STRESSES		
	X (m)	Y (m)	X (MPa)	Y (MPa)	XY (MPa)
133	.850	10.625	.2525E+00	-.5802E+00	.4845E+00

 ***** STOREY NO: 6 *****

Storey drift = 10.181 mm

GEOMETRY:

Wall 1 width = 6.000 m Wall 2 width = 5.100 m
 Beam length = 2.000 m Beam height = 1.000 m
 Storey height = 3.500 m Thickness = .200 m

LOADS:

Lateral top LDL = .000 kN/m Bot. LDL = .000 kN/m
 Conc. lat. load = 380.000 kN Vert. UDL = 70.000 kN/m

FORCES IN CONNECTING BEAM:

	At Wall 1	At midspan	At Wall 2
Axial (kN)	243.074	243.074	-243.074
Shear (kN)	-43.825	113.825	183.825
Moment (kN*m)	-90.278	11.453	-137.372

MAXIMUM STRESS SUMMARY (in MPa):

X	Y	XY	ELEMENT
-.3303E+01			93
	-.9444E+00		160
		.1112E+01	93

WALL 1:

ELEMENT	LOCATION		STRESSES		
	X (m)	Y (m)	X (MPa)	Y (MPa)	XY (MPa)
81	.750	16.625	-.7625E+00	-.5715E-01	-.4066E+00
82	2.250	16.625	-.2414E+00	-.4414E+00	.1517E+00
83	3.750	16.625	-.1920E+00	-.6151E+00	.3109E+00
84	5.250	16.625	-.3283E+00	-.4267E+00	.4005E+00

WALL 2:

ELEMENT	LOCATION		STRESSES		
	X (m)	Y (m)	X (MPa)	Y (MPa)	XY (MPa)
157	.850	16.625	.1441E+00	-.6203E+00	.3134E+00

Do you wish to display data at different increments (Y/N)?
?Y

Do you want all data printed (Y/N)?
?N

After the following prompts, enter the starting value, end value, and increment for the information you want processed.

Structure has 6 storeys.
Enter start, end, and increment for storeys processed:
?3,3,1

Each storey has 4 elements in vertical direction.
Enter start, end, and increment for elements processed:
?1,4,3

Wall 1 has 4 elements in horizontal direction.
Enter start, end, and increment for elements processed:
?1,4,3

Wall 2 has 3 elements in horizontal direction.
Enter start, end, and increment for elements processed:
?1,3,2

**** STOREY NO: 3 **** Storey drift = 3.991 mm

GEOMETRY:
Wall 1 width = 6.000 m Wall 2 width = 5.100 m
Beam length = 3.000 m Beam height = .500 m
Storey height = 3.000 m Thickness = .200 m

LOADS:
Lateral top LDL = .000 kN/m Bot. LDL = .000 kN/m
Conc. lat. load = 180,000 kN Vert. UDL = 70,000 kN/m

FORCES IN CONNECTING BEAM:

	At Wall 1	At midspan	At Wall 2
Axial (kN)	53,474	53,474	-53,474
Shear (kN)	16,863	53,137	123,137
Moment (kN*m)	-30,154	12,017	-76,120

MAXIMUM STRESS SUMMARY (In MPa):

X	Y	XY	ELEMENT
- .6181E+00			45
	- .4703E+01		36
		.7826E+00	39

WALL 1:

ELEMENT	LOCATION		STRESSES		
	X (m)	Y (m)	X (MPa)	Y (MPa)	XY (MPa)
33	.750	7.625	-.3821E+00	.1048E+01	.1484E+00
36	5.250	7.625	-.2235E+00	-.4703E+01	.5256E+00
45	.750	9.875	-.6181E+00	.2019E+00	.6432E+00
48	5.250	9.875	-.8175E-01	-.1859E+01	.1012E+00

WALL 2:

ELEMENT	LOCATION		STRESSES		
	X (m)	Y (m)	X (MPa)	Y (MPa)	XY (MPa)
121	.850	7.625	.2504E+00	-.2880E+00	.5399E+00
123	4.250	7.625	.1567E-01	-.4096E+01	.3191E+00
130	.850	9.875	.1282E+00	-.7929E+00	-.2752E-01
132	4.250	9.875	-.3439E-01	-.3332E+01	.2731E+00

Do you wish to display data at different increments (Y/N)?
 ?N

Processed data have been written on file CPI

Do you want to process different data (Y/N)?
 ?N

++++ NORMAL TERMINATION +++++

STOP

Appendix H

Listing of file created by preprocessor and used and modified by postprocessor

GEOMETRY DATA

CONCRETE PARAMETERS:

1) Density = 2400. kg/cu.m; 2) Strength = 30.0 MPa; 3) E = 27691. MPa

WALL WIDTHS:

4) Wall 1 = 6.000 m; 5) Gap = 2.000 m; 6) Wall 2 = 5.100 m

7) Number of storeys = 6

8) No. of elements in horizontal direction in Wall 1 = 4

9) Maximum permissible aspect ratio = 10.0

STOREY WALL THICKNESS STOREY HEIGHT BEAM HEIGHT

	(m)	(m)	(m)
1	.200	4.000	1.500
2	.200	3.000	.500
3	.200	3.000	.500
4	.200	3.000	.500
5	.200	3.000	.500
6	.200	3.500	1.000

APPLIED LOADS:

STOREY VERTICAL UDL LATERAL CONC. LOAD

	(kN/m)	(kN)
1	70.000	80.000
2	70.000	150.000
3	70.000	180.000
4	70.000	240.000
5	70.000	340.000
6	70.000	380.000

LINEARLY VARYING LATERAL LOAD (kN/m):

7) Bottom = .000; 8) Top = .000

Highest aspect ratio = 6.8

Max. allowable ratio = 10.0

No. of elements in horizontal direction in Wall 1 = 4

No. of elements in horizontal direction in Wall 2 = 3

No. of elements in vertical direction per storey = 4

No. of nodes: 225

No. of elastic beams: 6

No. of rigid beams: 24

No. of quad. plates: 168

RAW DATA:

	6	168	126	4	3	2
	6.000	2.000	5.100	.000	.000	.000
1	.200	2.500	1.500	70.000	80.000	
2	.200	2.500	.500	70.000	150.000	
3	.200	2.500	.500	70.000	180.000	
4	.200	2.500	.500	70.000	240.000	
5	.200	2.500	.500	70.000	340.000	
6	.200	2.500	1.000	70.000	380.000	

END

 PROCESSED DATA

 ***** STOREY NO: 1 *****

Storey drift = .975 mm

GEOMETRY:

Wall 1 width = 6.000 m Wall 2 width = 5.100 m
 Beam length = 2.000 m Beam height = 1.500 m
 Storey height = 4.000 m Thickness = .200 m

LOADS:

Lateral top LDL = .000 kN/m Bot. LDL = .000 kN/m
 Conc. lat. load = 80.000 kN Vert. UDL = 70.000 kN/m

FORCES IN CONNECTING BEAM:

	At Wall 1	At midspan	At Wall 2
Axial (kN)	187.767	187.767	-187.767
Shear (kN)	-8.630	78.630	148.630
Moment (kN*m)	-55.313	11.683	-101.947

MAXIMUM STRESS SUMMARY (in MPa):

X	Y	XY	ELEMENT
-.8167E+00			4
	-.8819E+01		4
		.1140E+01	7

WALL 1:

ELEMENT	LOCATION		STRESSES		
	X (m)	Y (m)	X (MPa)	Y (MPa)	XY (MPa)
1	.750	.625	.2511E+00	.3187E+01	.5089E+00
2	2.250	.625	-.4069E-01	-.5916E+00	.6561E+00
3	3.750	.625	-.6495E+00	-.4069E+01	.5468E+00
4	5.250	.625	-.8167E+00	-.8819E+01	.4193E+00
5	.750	1.875	-.2681E-01	.2639E+01	.3938E+00
6	2.250	1.875	.2959E-01	-.7649E+00	.8957E+00
7	3.750	1.875	-.2100E+00	-.3632E+01	.1140E+01
8	5.250	1.875	-.1942E+00	-.8416E+01	-.2981E+00
9	.750	2.875	.6981E-02	.2174E+01	.4159E+00
10	2.250	2.875	-.6132E-01	-.8300E+00	.8224E+00
11	3.750	2.875	.4716E+00	-.4298E+01	.1128E+01
12	5.250	2.875	.3723E+00	-.3174E+01	-.2579E+00
13	.750	3.625	-.1822E+00	.1822E+01	.4568E+00
14	2.250	3.625	-.6819E-01	-.8882E+00	.8489E+00
15	3.750	3.625	.4079E+00	-.4092E+01	.4147E+00
16	5.250	3.625	.1773E+00	-.2991E+01	.8136E+00

WALL 2:

ELEMENT	LOCATION		STRESSES		
	X (m)	Y (m)	X (MPa)	Y (MPa)	XY (MPa)
97	.850	.625	.9876E-01	.1125E+01	.4986E+00
98	2.550	.625	-.4013E+00	-.3369E+01	.7735E+00
99	4.250	.625	-.6496E+00	-.7883E+01	.8770E+00
100	.850	1.875	-.3003E+00	.5222E+00	.6054E+00
101	2.550	1.875	-.2843E+00	-.3439E+01	.1044E+01
102	4.250	1.875	-.6089E-01	-.7122E+01	.4995E+00
103	.850	2.875	-.6589E+00	-.6958E+00	.4995E+00
104	2.550	2.875	-.2636E+00	-.3172E+01	.8842E+00
105	4.250	2.875	-.3708E-01	-.6634E+01	.4228E+00
106	.850	3.625	-.3485E+00	-.7930E+00	.4395E+00
107	2.550	3.625	-.2286E+00	-.3191E+01	.7056E+00
108	4.250	3.625	-.5679E-01	-.6291E+01	.3945E+00

 ***** STOREY NO: 2 *****

Storey drift = 2.298 mm

GEOMETRY:

Wall 1 width = 6.000 m Wall 2 width = 5.100 m
 Beam length = 2.000 m Beam height = .500 m
 Storey height = 3.000 m Thickness = .200 m

LOADS:

Lateral top LDL = .000 kN/m Bot. LDL = .000 kN/m
 Conc. lat. load = 150.000 kN Vert. UDL = 70.000 kN/m

FORCES IN CONNECTING BEAM:

	At Wall 1	At midspan	At Wall 2
Axial (kN)	36.933	36.933	-36.933
Shear (kN)	26.656	43.344	113.344
Moment (kN*m)	-20.359	12.015	-66.330

MAXIMUM STRESS SUMMARY (in MPa):

X	Y	XY	ELEMENT
-.5167E+00			29
	-.6896E+01		20
		.1027E+01	20

WALL 1:

ELEMENT	LOCATION		STRESSES		
	X (m)	Y (m)	X (MPa)	Y (MPa)	XY (MPa)
17	.750	4.625	-.1631E+00	.1797E+01	.2495E+00
18	2.250	4.625	-.1339E-01	-.7239E+00	.7254E+00
19	3.750	4.625	-.2068E+00	-.2697E+01	.4881E+00
20	5.250	4.625	-.4162E+00	-.6896E+01	.1027E+01
21	.750	5.875	-.8707E-01	.1344E+01	.3694E+00
22	2.250	5.875	-.2251E+00	-.7958E+00	.6848E+00
23	3.750	5.875	-.3867E+00	-.2868E+01	.9818E+00
24	5.250	5.875	-.1986E+00	-.6083E+01	.4543E+00
25	.750	6.625	-.1806E+00	.1034E+01	.4986E+00
26	2.250	6.625	-.4557E+00	-.7479E+00	.7100E+00
27	3.750	6.625	-.1010E+00	-.3574E+01	.9651E+00
28	5.250	6.625	.2918E+00	-.2771E+01	.2433E+00

29	.750	6.875	-.5167E+00	.9124E+00	.5926E+00
30	2.250	6.875	-.3804E+00	-.7574E+00	.6808E+00
31	3.750	6.875	-.5021E-01	-.3526E+01	.8423E+00
32	5.250	6.875	.1090E+00	-.2691E+01	.2995E+00

WALL 2:

ELEMENT	LOCATION		STRESSES		
	X (m)	Y (m)	X (MPa)	Y (MPa)	XY (MPa)
109	.850	4.625	.1098E+00	.1721E+00	.6009E+00
110	2.550	4.625	-.8300E-01	-.2989E+01	.6559E+00
111	4.250	4.625	-.2532E-01	-.5520E+01	.3401E+00
112	.850	5.875	-.2630E+00	-.4081E+00	.6183E+00
113	2.550	5.875	.4189E-01	-.2740E+01	.6556E+00
114	4.250	5.875	-.1163E-01	-.5100E+01	.3229E+00
115	.850	6.625	-.4068E+00	-.7472E+00	.1196E+00
116	2.550	6.625	.4379E-01	-.2619E+01	.7530E+00
117	4.250	6.625	-.1494E-01	-.4841E+01	.3010E+00
118	.850	6.875	.9554E-01	-.7804E+00	.8143E-01
119	2.550	6.875	-.8962E-01	-.2667E+01	.7517E+00
120	4.250	6.875	-.1229E-01	-.4730E+01	.2979E+00

***** STOREY NO: 3 *****

Storey drift = 3.991 mm

GEOMETRY:

Wall 1 width = 6.000 m Wall 2 width = 5.100 m
Beam length = 2.000 m Beam height = .500 m
Storey height = 3.000 m Thickness = .200 m

LOADS:

Lateral top UDL = .000 kN/m Bot. UDL = .000 kN/m
Conc. lat. load = 180.000 kN Vert. UDL = 70.000 kN/m

FORCES IN CONNECTING BEAM:

	At Wall 1	At midspan	At Wall 2
Axial (kN)	53.474	53.474	-53.474
Shear (kN)	16.863	53.137	123.137
Moment (kN*m)	-30.154	12.017	-76.120

MAXIMUM STRESS SUMMARY (in MPa):

X	Y	XY	ELEMENT
-.6181E+00			45
	-.4703E+01		36
		.7826E+00	39

WALL 1:

ELEMENT	LOCATION		STRESSES		
	X (m)	Y (m)	X (MPa)	Y (MPa)	XY (MPa)
33	.750	7.625	-.3821E+00	.1048E+01	.1484E+00
34	2.250	7.625	-.1833E+00	-.6967E+00	.7661E+00
35	3.750	7.625	-.1636E+00	-.2389E+01	.6733E+00

36	5.250	7.625	-.2235E+00	-.4703E+01	.5256E+00
37	.750	8.875	-.8882E-01	.6345E+00	.3770E+00
38	2.250	8.875	-.2513E+00	-.8818E+00	.6747E+00
39	3.750	8.875	-.2316E+00	-.2249E+01	.7826E+00
40	5.250	8.875	-.4108E-01	-.4127E+01	.2791E+00
41	.750	9.625	-.2339E+00	.3233E+00	.5236E+00
42	2.250	9.625	-.3869E+00	-.8292E+00	.6520E+00
43	3.750	9.625	-.8328E-01	-.2623E+01	.7353E+00
44	5.250	9.625	.1911E+00	-.1928E+01	.9131E-01
45	.750	9.875	-.6181E+00	.2019E+00	.6432E+00
46	2.250	9.875	-.3198E+00	-.8125E+00	.5486E+00
47	3.750	9.875	-.3367E-01	-.2567E+01	.7078E+00
48	5.250	9.875	-.8175E-01	-.1859E+01	.1012E+00

WALL 2:

ELEMENT	LOCATION		STRESSES		
	X (m)	Y (m)	X (MPa)	Y (MPa)	XY (MPa)
121	.850	7.625	.2504E+00	-.2880E+00	.5399E+00
122	2.550	7.625	-.8105E-01	-.2351E+01	.6292E+00
123	4.250	7.625	.1567E-01	-.4096E+01	.3191E+00
124	.850	8.875	-.3690E+00	-.8267E+00	.5845E+00
125	2.550	8.875	.2542E-01	-.2119E+01	.5893E+00
126	4.250	8.875	-.2041E-01	-.3701E+01	.3144E+00
127	.850	9.625	-.4391E+00	-.7610E+00	.1119E-01
128	2.550	9.625	.1889E-01	-.2114E+01	.6884E+00
129	4.250	9.625	-.3539E-01	-.3438E+01	.2836E+00
130	.850	9.875	.1282E+00	-.7929E+00	-.2752E-01
131	2.550	9.875	-.1202E+00	-.2171E+01	.6948E+00
132	4.250	9.875	-.3439E-01	-.3332E+01	.2731E+00

**** STOREY NO: 4 ****

Storey drift = 5.901 mm

GEOMETRY:

Wall 1 width = 6.000 m Wall 2 width = 5.100 m
Beam length = 2.000 m Beam height = .500 m
Storey height = 3.000 m Thickness = .200 m

LOADS:

Lateral top LDL = .000 kN/m Bot. LDL = .000 kN/m
Conc. lat. load = 240.000 kN Vert. UDL = 70.000 kN/m

FORCES IN CONNECTING BEAM:

	At Wall 1	At midspan	At Wall 2
Axial (kN)	81.261	81.261	-81.261
Shear (kN)	11.766	58.234	128.234
Moment (kNm)	-35.273	12.039	-81.195

MAXIMUM STRESS-SUMMARY (in MPa):

X	Y	XY	ELEMENT
-.8394E+00			61
	-.3031E+01		52
		.6879E+00	61

WALL 1:

=====

ELEMENT	LOCATION		STRESSES		
	X (m)	Y (m)	X (MPa)	Y (MPa)	XY (MPa)
49	.750	10.625	-.4355E+00	.3751E+00	.8090E-01
50	2.250	10.625	-.1764E+00	-.6719E+00	.6426E+00
51	3.750	10.625	-.1417E+00	-.1667E+01	.5372E+00
52	5.250	10.625	-.2456E+00	-.3031E+01	.4310E+00
53	.750	11.875	-.1034E+00	.4357E-01	.3352E+00
54	2.250	11.875	-.3047E+00	-.8080E+00	.5450E+00
55	3.750	11.875	-.2565E+00	-.1542E+01	.5925E+00
56	5.250	11.875	-.2745E-01	-.2571E+01	.2190E+00
57	.750	12.625	-.3257E+00	-.2186E+00	.5187E+00
58	2.250	12.625	-.4540E+00	-.7545E+00	.5109E+00
59	3.750	12.625	-.1527E+00	-.1728E+01	.5403E+00
60	5.250	12.625	.2057E-01	-.1219E+01	.1981E-01
61	.750	12.875	-.8394E+00	-.3314E+00	.6879E+00
62	2.250	12.875	-.3611E+00	-.7114E+00	.3267E+00
63	3.750	12.875	-.1025E+00	-.1688E+01	.6007E+00
64	5.250	12.875	-.2919E+00	-.1156E+01	-.2115E-01

WALL 2:

=====

ELEMENT	LOCATION		STRESSES		
	X (m)	Y (m)	X (MPa)	Y (MPa)	XY (MPa)
133	.850	10.625	.2525E+00	-.5802E+00	.5845E+00
134	2.550	10.625	-.1168E+00	-.1786E+01	.5625E+00
135	4.250	10.625	.8305E-02	-.2738E+01	.2839E+00
136	.850	11.875	-.4336E+00	-.1069E+01	.5672E+00
137	2.550	11.875	.6618E-02	-.1550E+01	.4921E+00
138	4.250	11.875	-.2637E-01	-.2397E+01	.2717E+00
139	.850	12.625	-.4949E+00	-.7213E+00	-.6693E-01
140	2.550	12.625	.9858E-02	-.1603E+01	.5926E+00
141	4.250	12.625	-.4658E-01	-.2161E+01	.2349E+00
142	.850	12.875	.1069E+00	-.7480E+00	-.1165E+00
143	2.550	12.875	-.1365E+00	-.1663E+01	.6023E+00
144	4.250	12.875	-.4388E-01	-.2067E+01	.2211E+00

 ***** STOREY NO: 5 *****

Storey drift = 7.888 mm

GEOMETRY:

Wall 1 width = 6.000 m Wall 2 width = 5.100 m
 Beam length = 2.000 m Beam height = .500 m
 Storey height = 3.000 m Thickness = .200 m

LOADS:

Lateral top LDL = .000 kN/m Bot. LDL = .000 kN/m
 Conc. lat. load = 340.000 kN Vert. UDL = 70.000 kN/m

FORCES IN CONNECTING BEAM:

	At Wall 1	At midspan	At Wall 2
Axial (kN)	128.176	128.176	-128.176
Shear (kN)	10.317	59.684	129.683
Moment (kN*m)	-36.763	12.079	-82.604

MAXIMUM STRESS SUMMARY (In MPa):

X Y XY ELEMENT
 -.1176E+01
 -.1644E+01
 .7173E+00
 77
 6E
 77

WALL 1:

ELEMENT	LOCATION		STRESSES		
	X (m)	Y (m)	X (MPa)	Y (MPa)	XY (MPa)
65	.750	13.625	-.5724E+00	-.7042E-01	-.7409E-01
66	2.250	13.625	-.2258E+00	-.5366E+00	.4599E+00
67	3.750	13.625	-.1819E+00	-.1014E+01	.4060E+00
68	5.250	13.625	-.2784E+00	-.1644E+01	.3707E+00
69	.750	14.875	-.1208E+00	-.2403E+00	.2464E+00
70	2.250	14.875	-.3761E+00	-.6802E+00	.3396E+00
71	3.750	14.875	-.2786E+00	-.9481E+00	.4111E+00
72	5.250	14.875	-.4754E-01	-.1279E+01	.1654E+00
73	.750	15.625	-.4436E+00	-.4143E+00	.4793E+00
74	2.250	15.625	-.5213E+00	-.6389E+00	.2757E+00
75	3.750	15.625	-.1728E+00	-.1011E+01	.3452E+00
76	5.250	15.625	-.1790E+00	-.6442E+00	.2641E-01
77	.750	15.875	-.1176E+01	-.5069E+00	.7173E+00
78	2.250	15.875	-.3643E+00	-.5673E+00	-.3110E-01
79	3.750	15.875	-.1269E+00	-.1012E+01	.5209E+00
80	5.250	15.875	-.4920E+00	-.5868E+00	-.7172E-01

WALL 2:

ELEMENT	LOCATION		STRESSES		
	X (m)	Y (m)	X (MPa)	Y (MPa)	XY (MPa)
145	.850	13.625	.2325E+00	-.7041E+00	.4048E+00
146	2.550	13.625	-.1482E+00	-.1231E+01	.4624E+00
147	4.250	13.625	.2254E-02	-.1523E+01	.2248E+00
148	.850	14.875	-.5085E+00	-.1119E+01	.5348E+00
149	2.550	14.875	-.3871E-01	-.9867E+00	.3577E+00
150	4.250	14.875	-.4116E-01	-.1264E+01	.1994E+00
151	.850	15.625	-.6155E+00	-.5997E+00	.1396E+00
152	2.550	15.625	-.4542E-01	-.1057E+01	.4471E+00
153	4.250	15.625	-.6765E-01	-.1080E+01	.1478E+00
154	.850	15.875	-.7125E-02	-.6164E+00	-.1988E+00
155	2.550	15.875	-.2025E+00	-.1113E+01	.4504E+00
156	4.250	15.875	-.6407E-01	-.1008E+01	.1281E+00

 ***** STOREY NO: 6 *****

Storey drift = 10.181 mm

GEOMETRY:

Wall 1 width = 6.000 m Wall 2 width = 5.100 m
 Beam length = 2.000 m Beam height = 1.000 m
 Storey height = 3.500 m Thickness = .200 m

LOADS:

Lateral top LDL = .000 kN/m Bot. LDL = .000 kN/m
 Conc. lat. load = 380.000 kN Vert. UDL = 70.000 kN/m

FORCES IN CONNECTION BEAM

	AT WALL 1	AT MIDSPAN	AT WALL 2
Axial (kN)	243.074	243.074	-243.074
Shear (kN)	-43.625	113.625	183.625
Moment (kN*m)	-9.275	11.453	-137.372

MAXIMUM STRESS SUMMARY (in MPa)

X	Y	XY	ELEMENT
-0.3303E+01			93
	-0.9444E+00		161
		0.1112E+01	93

WALL 1:

=====

ELEMENT	LOCATION		STRESSES		
	X (m)	Y (m)	X (MPa)	Y (MPa)	XY (MPa)
81	0.750	16.625	-0.7625E+00	-0.5715E-01	-0.4066E+00
82	2.250	16.625	-0.2414E+00	-0.4414E+00	0.1517E+00
83	3.750	16.625	-0.1920E+00	-0.6151E+00	0.3109E+00
84	5.250	16.625	-0.3283E+00	-0.4267E+00	0.4005E+00
85	0.750	17.875	-0.7099E-01	0.8556E-01	-0.7699E-01
86	2.250	17.875	-0.6096E+00	-0.7372E+00	0.2025E+00
87	3.750	17.875	-0.6668E+00	-0.6607E+00	0.1584E+00
88	5.250	17.875	-0.4362E-01	-0.3104E+00	0.1724E+00
89	0.750	18.750	-0.3895E+00	0.1760E-01	0.2029E+00
90	2.250	18.750	-0.1458E+01	0.7953E+00	0.6058E+00
91	3.750	18.750	-0.9660E+00	-0.4095E+00	-0.2017E+00
92	5.250	18.750	-0.7350E+00	-0.3955E+00	-0.2533E+00
93	0.750	19.250	-0.3303E+01	-0.2552E+00	0.1112E+01
94	2.250	19.250	-0.1440E+01	-0.4919E+00	-0.9748E-01
95	3.750	19.250	-0.1208E+01	-0.3142E+00	0.6379E-01
96	5.250	19.250	-0.1888E+01	-0.4249E+00	-0.3105E+00

WALL 2:

=====

ELEMENT	LOCATION		STRESSES		
	X (m)	Y (m)	X (MPa)	Y (MPa)	XY (MPa)
157	0.850	16.625	0.1441E+00	-0.6203E+00	0.3134E+00
158	2.550	16.625	-0.2356E+00	-0.6415E+00	0.2973E+00
159	4.250	16.625	-0.3156E-01	-0.5458E+00	0.1043E+00
160	0.850	17.875	-0.5995E+00	-0.9444E+00	0.5542E+00
161	2.550	17.875	-0.1480E+00	-0.3234E+00	0.1195E+00
162	4.250	17.875	-0.3664E-01	-0.4515E+00	0.4122E-01
163	0.850	18.750	-0.6437E+00	-0.2366E+00	-0.6222E-01
164	2.550	18.750	-0.8901E-01	-0.3516E+00	0.5457E-01
165	4.250	18.750	-0.1697E-01	-0.3802E+00	0.1414E-01
166	0.850	19.250	0.1292E+00	-0.1925E+00	-0.1243E+00
167	2.550	19.250	-0.9612E-01	-0.3899E+00	0.2104E-01
168	4.250	19.250	-0.1426E-01	-0.3500E+00	0.3618E-02

**** STOREY NO. 2 **** Storey drift = 2.298 mm

GEOMETRY:

Wall 1 width = 6.000 m Wall 2 width = 5.100 m
 Beam length = 2.000 m Beam height = .500 m
 Storey height = 3.000 m Thickness = .200 m

LOADS:

Lateral top LDL = .000 kN/m Bot. LDL = .000 kN/m
 Conc. lat. load = 150.000 kN Vert. UDL = 70.000 kN/m

FORCES IN CONNECTING BEAM:

	At Wall 1	At midspan	At Wall 2
Axial (kN)	36.933	36.933	-36.933
Shear (kN)	26.656	43.344	113.544
Moment (kN*m)	-20.359	12.015	-66.530

MAXIMUM STRESS SUMMARY (in MPa):

X	Y	XY	ELEMENT
-.5167E+00			29
	-.6896E+01		20
		.1027E+01	20

WALL 1:
 =====

ELEMENT	LOCATION		STRESSES		
	X (m)	Y (m)	X (MPa)	Y (MPa)	XY (MPa)
17	.750	4.625	-.1631E+00	.1797E+01	.2495E+00
18	2.250	4.625	-.1339E-01	-.7239E+00	.7254E+00
19	3.750	4.625	-.2068E+00	-.2697E+01	.4881E+00
20	5.250	4.625	-.4162E+00	-.6896E+01	.1027E+01

WALL 2:
 =====

ELEMENT	LOCATION		STRESSES		
	X (m)	Y (m)	X (MPa)	Y (MPa)	XY (MPa)
109	.850	4.625	.1098E+00	.1721E+00	.6009E+00

**** STOREY NO: 4 **** Storey drift = 5.901 mm

GEOMETRY:

Wall 1 width = 6.000 m Wall 2 width = 5.100 m
 Beam length = 2.000 m Beam height = .500 m
 Storey height = 3.000 m Thickness = .200 m

LOADS:

Lateral top LDL = .000 kN/m Bot. LDL = .000 kN/m
 Conc. lat. load = 240.000 kN Vert. UDL = 70.000 kN/m

FORCES IN CONNECTING BEAM:

	At Wall 1	At midspan	At Wall 2
Axial (kN)	81.261	81.261	-81.261
Shear (kN)	11.766	58.234	128.234
Moment (kN*m)	-35.273	12.039	-81.195

MAXIMUM STRESS SUMMARY (in MPa):

X	Y	XY	ELEMENT
-.8394E+00			61
	-.3031E+01		52
		.6879E+00	61

WALL 1:

ELEMENT	LOCATION		STRESSES		
	X (m)	Y (m)	X (MPa)	Y (MPa)	XY (MPa)
49	.750	10.625	-.4355E+00	.3751E+00	.8090E-01
50	2.250	10.625	-.1764E+00	-.6719E+00	.6426E+00
51	3.750	10.625	-.1417E+00	-.1667E+01	.5372E+00
52	5.250	10.625	-.2456E+00	-.3031E+01	.4310E+00

WALL 2:

ELEMENT	LOCATION		STRESSES		
	X (m)	Y (m)	X (MPa)	Y (MPa)	XY (MPa)
133	.850	10.625	.2525E+00	-.5802E+00	.4845E+00

 ***** STOREY NO: 6 *****

Storey drift = 10.181 mm

GEOMETRY:

Wall 1 width = 6.000 m Wall 2 width = 5.100 m
 Beam length = 2.000 m Beam height = 1.000 m
 Storey height = 3.500 m *Thickness = .200 m

LOADS:

Lateral top LDL = .000 kN/m Bot. LDL = .000 kN/m
 Conc. lat. load = 380.000 kN Vert. UDL = 70.000 kN/m

FORCES IN CONNECTING BEAM:

	At Wall 1	At midspan	At Wall 2
Axial (kN)	243.074	243.074	-243.074
Shear (kN)	-43.825	113.825	183.825
Moment (kN*m)	-90.278	11.453	-137.372

MAXIMUM STRESS SUMMARY (in MPa):

X	Y	XY	ELEMENT
-.3303E+01			93
	-.9444E+00		160
		.1112E+01	93

WALL 1:

ELEMENT	LOCATION		STRESSES		
	X (m)	Y (m)	X (MPa)	Y (MPa)	XY (MPa)
81	.750	16.625	-.7625E+00	-.5715E-01	-.4066E+00
82	2.250	16.625	-.2414E+00	-.4414E+00	.1517E+00
83	3.750	16.625	-.1920E+00	-.6151E+00	.3109E+00
84	5.250	16.625	-.3283E+00	-.4267E+00	.4005E+00

WALL 2:

ELEMENT	LOCATION		STRESSES		
	X (m)	Y (m)	X (MPa)	Y (MPa)	XY (MPa)
157	.850	16.625	.1441E+00	-.6203E+00	.3134E+00

 ***** STOREY NO: 3 *****

Storey drift = .3,991 mm

GEOMETRY:

Wall 1 width = 6.000 m Wall 2 width = 5.100 m
 Beam length = 2.000 m Beam height = .500 m
 Storey height = 3.000 m Thickness = .200 m

LOADS:

Lateral top LDL = .000 kN/m Bot. LDL = .000 kN/m
 Conc. lat. load = 180.000 kN Vert. UDL = 70.000 kN/m

FORCES IN CONNECTING BEAM:

	At Wall 1	At midspan	At Wall 2
Axial (kN)	53.474	53.474	-53.474
Shear (kN)	16.863	53.137	123.137
Moment (kN*m)	-30.154	12.017	-76.120

MAXIMUM STRESS SUMMARY (in MPa):

X	Y	XY	ELEMENT
-.6181E+00			45
	-.4703E+01		36
		.7826E+00	39

WALL 1:

ELEMENT	LOCATION		STRESSES		
	X (m)	Y (m)	X (MPa)	Y (MPa)	XY (MPa)
33	.750	7.625	-.3821E+00	.1048E+01	.1484E+00
36	5.250	7.625	-.2235E+00	-.4703E+01	.5256E+00
45	.750	9.875	-.6181E+00	.2019E+00	.6432E+00

48 5.250 9.875 -.8175E-01 -.1859E+01 .1012E+00

WALL 2:

ELEMENT	LOCATION		STRESSES		
	X (m)	Y (m)	X (MPa)	Y (MPa)	XY (MPa)
121	.850	7.625	.2504E+00	-.2880E+00	.5399E+00
123	4.250	7.625	.1567E-01	-.4096E+01	.3191E+00
130	.850	9.875	.1282E+00	-.7929E+00	-.2752E-01
132	4.250	9.875	-.3439E-01	-.3332E+01	.2731E+00

END

1 2 4 0 6 . 8 6

FIN