COMPUTER-AIDED DESIGN OF REINFORCED CONCRETE SHALLOW FOUNDATIONS FOR FRAMED BUILDINGS

BY

HASSAN, KULIYA

A thesis submitted to the Post Graduate School, Ahmadu Bello University, Zaria, in partial fulfillment of the requirements for the award of Master of Science Degree in Structural Engineering.

Department of Civil Engineering,
Ahmadu Bello University,
5iaria.

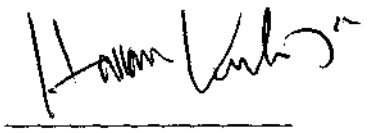
OCTOBER, 199 8

DEDICATION

Dedicated to Mami, Fatima A., Zainab and IEA.

DECLARATION

I declare that this thesis is my own work and has not been previously submitted in any form for the award of any degree. All works, opinions and ideas other than mine have been fully acknowledged.



HASSAN, KULIYA

CERTIFICATION

This thesis entitled "Computer-Aided Design of Reinforced Concrete Shallow Foundations for Framed Buildings", by Hassan **Kuliya**, meets the regulations governing the award of the Degree of Master of Science [Structural Engineering] of Ahmadu Bello University, and is approved for its contribution to knowledge and literary presentation.

Date: 16-2-99

Dr. J. O. Afolayan
Chairman, Supervisory Committee

Date: 16.2.99

Dr. K. J. Osinubi
Head of Department

Date: 01-03-99

Dr. S. P. Ejeh
Member, Supervisory Committee

Date: 25/06/99

Dean, Postgraduate School

ACKNOWLEDGEMENTS

I acknowledge the scholarly and friendly guidance of my supervisor, Dr. J.O. Afolayan. His suggestions and advice were invaluable. Without his moral support, this work may not have been completed. The suggestions of Engr. Abejide O.S. also went a long way to make this work what it is; without his assistance, the work may not have been possible.

I wish to also thank Dr. K.J. Osinubi for his support and understanding when I was developing and test running the computer program. Likewise, the moral and brotherly encouragement of Dr. S.P. Ejeh can not escape mention.

Finally, I am sincerely grateful for the fatherly concern and support of Professor Adebisi since the start of my course-work through to the completion of this thesis.

ABSTRACT

A computer program for the design of reinforced concrete shallow foundations is presented. The choice of the type of foundation is made interactively with the computer based on experience in design decisions.

The computer program developed was used to design reinforced concrete foundations such as pads, combined, strip and raft footings that had been previously designed using manual methods in order to check its validity. The results obtained from the computer program turned out to be optimized versions of the manually computed results.

The following advantages of the program over manual design were observed:

- Elimination of tedious arithmetic and the attendant errors involved in manual calculations. This results in the accuracy of results
- High speed in the processing of data to produce a design
- Versatility in handling data
- Optimisation of designs

TABLE OF CONTENTS

CHAPTER		PAGE
	TITLE PAGE	i
	DEDICATION	ii
	DECLARATION	iii
	CERTIFICATION	iv
	ACKNOWLEDGEMENTS	v
	ABSTRACT	vi
	TABLE OF CONTENTS	viii
1.0	INTRODUCTION	1
1.1	General	1
1.2	Aims and Objectives	2
1.3	Scope and Limitations	3
2.0	LITERATURE REVIEW	5
3.0	PRINCIPLES OF ANALYSIS AND DESIGN OF REINFORCED CONCRETE SHALLOW FOUNDATIONS	9
3.1	General Principles of Foundation Design	9
3.2	Distribution of Soil Pressure	10
3.3	Section Analysis of Reinforced Concrete Shallow Foundation	10
3.3.1	Distribution of Strains and Stresses	11
3.3.2	Pad Foundation	13
3.3.3	Combined Foundations	14
3.3.4	Strip Foundations	17
3.3.5	Raft Foundations	18
4.0	DESIGN ALGORITHM AND PROGRAM DEVELOPMENT	20
4.1	Introduction	20
4.2	Program Logic	20

4.3	Program FOUND	21
4.3.1	Running the Program	21
4.3.2	Viewing Results	21
4.3.3	Program FOUND	21
4.3.4	Arrays	22
4.3.5	Subroutines	25
4.4	Parameter Definition of Program Found	28
4.5	Program Listing	28
4.6	Typical Desigr Examples of Shallow Foundations for Structures	29
4.6.1	Pad Foundation	29
4.6.2	Combined Footing	34
4.6.3	Strip Foundation	38
4.6.4	Raft Foundation	42
4.7	Comparison of Manual Results and Results from FOUND	46
4.8	Discussion of Results	48
4.8.1	Pad Foundation	48
4.8.2	Combined Foundation	49
4.8.3	Strip Foundation	49
4.8.4	Raft Foundation	50
5.0	**CONCLUSION AND RECOMMENDATIONS**	**52**

REFERENCES

APPENDICES

APPENDIX I - PARAMETER LISTING

APPENDIX II - FLOW CHART OF MAIN PROGRAM

APPENDIX III - SOURCE CODE

LIST OF FIGURES

FIGURE		PAGE
3.1	Stress-strain relationship in reinforced concrete	11
3.2	Critical section and perimeter for bending and punching shear in pad footings	13
3.3	Structural diagram of a combined footing	15
3.4	Bending moment and shear force diagram for a combined footing	36

4.2 Moment and shear coefficients for an equal span continuous beam 39

LIST OF TABLES

TABLE PAGE

4.1 Maximum spacing of reinforcement for no redistribution of moments 31

4.2 Ultimate shear values 32

4.3 Ultimate local bond stress 33

CHAPTER 1

INTRODUCTION

1.1 General

This work presents a modest response to the need for simplified methods of analyzing and designing reinforced concrete shallow foundations for framed buildings in normal ground. Since the design of reinforced concrete foundations depends on the knowledge and adequate estimation of the loads from the superstructure and the nature of the ground in which it is to be located, a thorough understanding of the analysis of the superstructure in response to applied vertical and lateral loads, whether dead or live, is fundamental to the development of a suitable criterion. It is the loads from the superstructure that are transferred to the foundations through columns, walls or piers. Although reinforced concrete is not a truly elastic material, yet some valid assumptions have been made which include the compatibility of the stress - strain curves for concrete and steel [CP 110, 1972; BS 8110, 1985].

Foundations are one of the vital parts in a structure. It is usually the duty of an experienced structural engineer to design foundations in a consulting office due to the adverse effects the foundation may have on the entire structure if it is not adequately designed. However, this duty is

sometimes shifted to junior engineers when the team leader is not around. These junior engineers may not have the requisite experience to handle such jobs, even though they may hold the required academic qualifications. Therefore, to ensure safety of designs and avoidance of costly mistakes at all times, it is important that this process be standardized and made within easy reach always.

1.2 Aims and Objectives

Following the day-to-day introduction of electronic digital computers in design offices and places where they can be easily utilized the necessity for analytical tools specifically directed towards its use became apparent. It is with this notion in mind that a computer program named **FOUND** has been developed to analyze and design reinforced concrete shallow foundations for framed buildings.

A principal objective of this work is to provide a simplified approach to the use of the computer method, so that its benefits may be made available to the designer, even if relatively inexperienced, without the need for performing complicated analysis in the design office, and ensuring safe designs at all times. One of the intentions is to be able to assess the effect of modest structural design changes without incurring computational and time costs of a complete re-analysis.

One of such changes is usually associated with revising structural properties of sections in the light of the results of preliminary analysis.

This work is concerned with situations in which the arrangement and magnitudes of the applied loads, and support or ground conditions and are known. In practice, the designer must first estimate the member sizes of the foundation, analyze the structure, and compare the calculated and allowable stresses. If certain values turn out greater than the allowable ones, or if the allowable ones are unacceptably higher than the calculated values, changes will be made in the assumed configuration of the section. It may be beneficial to repeat this cycle a number of times, until an optimum section or member that satisfies all conditions is obtained. It is useful therefore to develop procedures alternative to complete conventional reprocessing which can obtain the results for the modified structure immediately.

1.3 Scope and Limitations

The properties of reinforced concrete foundations, including elastic behavior, are discussed in chapter three having due regard to earlier work done by others. In a nutshell, the strength, suitability and efficiency of a reinforced concrete foundation, whether it is a pad, combined, strip or raft footing is the primary

purpose of this work. Raft footings considered are the solid slab type only. The computer program FOUND gives the design of an optimum section for the foundation, as well as volume of concrete, weight of reinforcement, and weight of reinforcement per cubic meter of concrete. Chapter four discusses the development of the program FOUND. The results of an example on the use of the program to design some typical reinforced concrete foundations are also presented and discussed.

The program is based on BS8110 and CP110 for the design of rectangular footings subjected to axial loads only. BS8110 is used because of the convenience it offers in the use of equations as an alternative to the use of tables in CP110, for example in determining ultimate shear values.

The columns carried by the footings must be rectangular on plan and arranged in an orthogonal grid. The locations of both the columns and the designed footings are given in the Cartesian coordinate system. A maximum of 99 columns can be considered at a time.

CHAPTER 2

LITERATURE REVIEW

Structural design problems can usually be formulated as corresponding optimization problems, which can be solved by applying mathematical programming methods [Jozwiak, 1986]. An example of a mathematical programming method is the use of a computer language such as FORTRAN, to easily solve a related engineering or scientific problem when it is required.

The use of a computer language when applied to solve problems used to be a cumbersome idea due to the long time required in computing [Jozwiak, 1986]. However, the advent of very efficient artificial intelligence machines, such as microcomputers, incorporating powerful and vast scientific memory and information processing abilities has made this idea an outdated one. The continuous progress in computer technology and development of numerical methods eliminate the constraints arising from the size of computer resources necessary for calculations, which was the main obstacle against the application of mathematical programming methods in structural design.

Winston [1984] noted that intelligent computers should be used to check design rules and recommendations, and to recall relevant precedent designs. This can be achieved through an intelligent

and experienced engineer or designer storing necessary and sufficient information or procedure in the computer, so that when such information is required, the machine will give the desired results without failure.

Winston [1986] again noted that the main purpose of application of artificial intelligence in engineering design is to develop computer programs which function better as tools for engineers and designers. Sandewall [1978] remarked that people usually learn by being taught and by making their own original inventions.

Although the idea of building a universal, self-learning system has been rejected, the concept of developing self-improving computer programs at a much lower level of generality is still considered as real by many scientists, who think that the ability to self-modify operations and change decisions, basing upon previous design results, is the most important feature of intelligent programs [Rooney and Smith, 1982].

The main purpose of application of computer-aided designs in structural optimization is to shorten calculations and processing time. In his submission, Jozwiak [1986] developed a computer-aided design program for the analysis and design of space and plane trusses of arbitrary shape in steel. Abejide [1994] also developed a computer program KOM22 for the design

of continuous pre-stressed concrete beams. He noted that one of the advantages of the program is that it eliminates or at least reduces to a considerable extent, the cumbersome mathematical involvement and decisions necessary for an optimum design of a continuous pre-stressed section.

Many packages have been presented [Falk, 1985] for designs in reinforced concrete that are simply computer implementations of manual design procedures. He stated that in addition to rapid calculation, the packages provide the user with planned and organized solution procedures, and compel them to follow those procedures. Among the packages presented by Falk [1985] is a package for the design of reinforced concrete foundations for only pad footings in the American Concrete Institute code.

Bensasson et al [1980] was of the opinion that since computers are coming into more general use in design offices, computer programs based on CP 110 [1972] code requirements should be used for design purposes, so as to save time in design. He stressed that computer-aided design should not only be used for complicated designs but that simple design procedures can also benefit from such an operation.

It is obvious therefore that a computer-aided design in line with Bensasson et al's [1980] suggestion is a good idea. In line with this notion, an

intelligent procedure and computer program FOUND for the design of reinforced concrete shallow foundations for framed structures to CP 110 [1972] and BS 8110 [1985] is presented in this work.

Results obtained from the computer program compare favorably well with manually computed ones, and the program FOUND considerably reduces the cumbersomeness associated with manual operations. Calculation and table-reading errors are also greatly reduced, if not totally eliminated.

CHAPTER 3

PRINCIPLES OF ANALYSIS AND DESIGN OF REINFORCED CONCRETE SHALLOW FOUNDATIONS

3.1 General Principles of Foundation Design

A foundation is required to transmit and distribute loads from the columns and walls of a superstructure to the ground. This must be done without exceeding the safe bearing capacity of the soil if failure or excessive settlement of the structure is to be avoided.

In designing foundations, the loading considered when calculating base areas in contact with the ground are those that apply to the serviceability limit state, and they are:

1. dead + imposed loads $= 1.0G_K + 1.0Q_K$
2. dead + wind loads $= 1.0G_K + 1.0W_K$
3. dead + imposed + wind loads

$$=1.0G_K+0.8Q_K+0.8W_K$$

When foundations are subjected to vertical and horizontal loads, the following rule applies:

$$\frac{V}{P_v} + \frac{H}{P_h} \leq 1.0 \qquad (3.1)$$

where

V = vertical load

H = horizontal load

P_V = allowable vertical load

P_H = allowable horizontal load

Calculations to determine the structural strength of the foundation, that is, base thickness and areas of reinforcement, are based on loading and resultant ground pressure at the ultimate limit state.

3.2 Distribution of Soil Pressure

For the purpose of calculating the pressure distribution under the footings, the following assumptions [Reynolds and Steedman, 1978] are made:

1. The soil acts as an elastic material
2. The footing has infinite rigidity

Since axial loads only are considered for the purposes of this work, the above assumptions will give a uniform distribution of pressure under the footing. It is therefore simply calculated as:

$$EP = \frac{LOAD}{BASE\ AREA} \qquad (3.2)$$

where EP = Earth pressure

3.3 Section Analysis of Reinforced Concrete Shallow Foundations

The fundamental principles on which the foundation section members are designed are:

1. Stresses and strains are related by material properties, in this case concrete and steel.
2. Distribution of strains must be compatible with the deformed shape of the section.
3. The resultant forces developed by the section

must balance the applied loads for static equilibrium.

3.3.1 Distribution of Strains and Stresses

An equivalent rectangular stress block is used to model the stress-strain relation for the section of a member in bending. This is shown in figure 3.1 below.

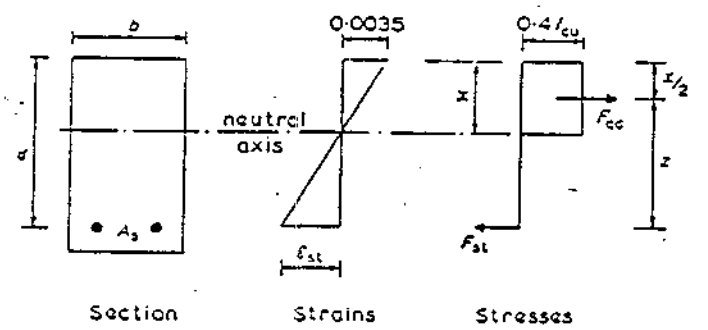


Figure 3.1 Stress - strain relationship in reinforced concrete.

The above distributions of strain and stress are based on the assumptions that (CP110, 1972):

1. Concrete in regions of tensile strains is cracked, and thereafter all tension is carried by the reinforcement
2. Plane sections remain plane after straining so that there is a linear distribution of strains

The simplified stress distribution above facilitates easier analysis of the section at the ultimate limit state. Thus, the basic design equations are derived as follows:

For equilibrium to be achieved on the section,

$$F_{cc} = F_{st} \tag{3.3}$$

Assuming the steel is at the yield stress,

$$F_{st} = 0.87 f_y A_s \tag{3.4}$$

where $0.87 = f_y/\gamma_m = f_y/1.15$

A_s = steel area provided as reinforcement

Other symbols are as in Fig. 3.1

To balance the couple created by the two forces F_{cc} and F_{st}, the section must develop an internal moment of resistance equal to

$$M_u = F_{cc} z = F_{st} z \tag{3.5}$$

$$\text{Therefore, } M_u = 0.87 f_y A_s z \tag{3.6}$$

$$\text{So that } A_s = M_u/(0.87 f_y z) \tag{3.7}$$

For equilibrium, the externally applied moment, M, on the foundation due to net upward pressure must be equal to the internal moment of resistance, M_u. Thus,

$$A_s = M/(0.87 f_y z) \tag{3.8}$$

To design the section of the foundation, the ultimate moment is calculated at a critical section, and reinforcement is provided based on this. The thickness of the base is determined by checking shears and ensuring that they do not exceed ultimate values. The critical sections for each type of foundation are discussed below.

3.3.2 Pad Foundations

For a single pad footing, the critical section for bending is at the column face [Reynolds and Steedman, 1978]. This is illustrated in figure 3.2 below:

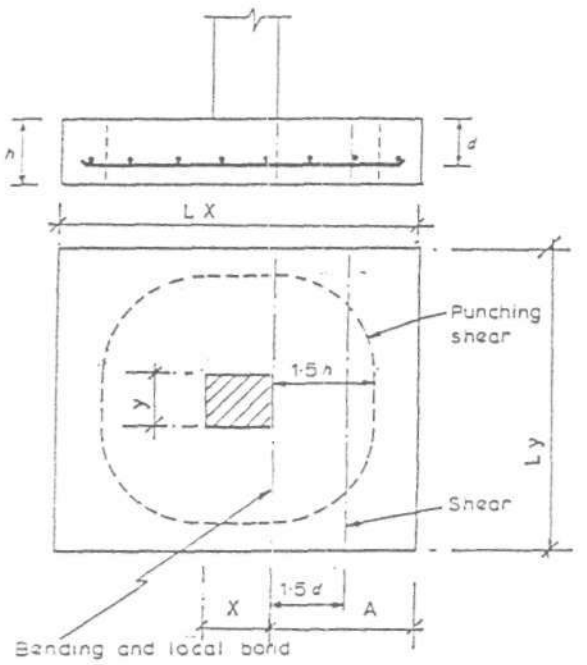


Figure 3.2 Critical section and perimeter for bending and punching shear in pad footings

$$a = 0.5L_x - 0.5x \quad (3.9)$$

$$= 0.5(L_x - x) \quad (3.10)$$

From statics, the moment M is:

$$M = w.\ L_y.\ a.\ a/2 \quad (3.11)$$

$$= 0.5w.L_y a^2 \quad (3.12)$$

where w = net upward pressure.

The critical section for punching shear is taken at a distance of 1.5h from the face of the column, where h is the overall depth of the footing. From

figure 3.2, the perimeter along the critical section is given by:

$$\text{Perimeter} = 2y + 2x + 4[1/4 \cdot \pi \cdot 2(1.5h)] \quad (3.13)$$

Thus,

$$\text{perimeter} = 2y + 2x + 3\pi h \quad (3.14)$$

The area within the perimeter (AWP) is given as

$$AWP = [2(1.5h)^2+x][2(1.5h) + y] + \pi(1.5h)^2 - 4[(1.5h)(1.5h)] \quad (3.15)$$

The shear force producing failure is the force outside this area. Therefore, the punching shear force, V, is given as

$$V = [L_x.L_y - AWP] .UPN$$

where UPN = net upward pressure.

3.3.3 Combined Foundations

Figure 3.3 indicates the structural diagram of a typical combined pad footing for two columns.

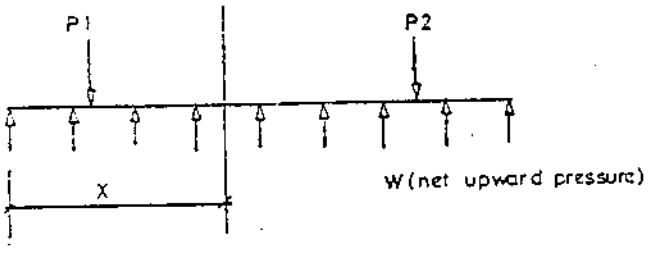


Figure 3.3 Structural diagram of a typical combined footing

The moment at any section at a distance x from the left edge is obtained by first solving for P_1 and P_2, the reactions due to the net upward pressure, by taking moments about P_2. The moment at any section x is then given by:

$$M_x = \frac{wx^2}{2} - P_2[x - b] - P_1[x - a] \qquad (3.16)$$

The point of maximum negative moment lies between P_1 and P_2, and is given by

$$\frac{dM_x}{dx} = 0 \qquad (3.17)$$

Differentiating by Macauly's method, the bracket gives

$$\frac{dM_x}{dx} = w.x - P_2 - P_1 = 0 \qquad (3.18)$$

which implies that

$$x = \frac{P_2 + P_1}{w} \tag{3.19}$$

Substituting this value of x into equation (3.16) gives the value of the negative moment.

Punching shear is not normally checked for combined footings because the critical perimeter mostly falls outside the base. However, in the case where it falls within the base, the calculation is the same as above.

Wide beam shear governs the design for thickness and it is determined in the usual manner by summing up forces to one side of the section at which shear is checked. This is normally at the point of contra-flexure, which is obtained by setting the moment equation (equation 3.16) to zero, and solving the resulting quadratic equation.

Thus,

$$M = \frac{wx^2}{2} - P_2[x-b] - P_1[x-a] = 0 \tag{3.20}$$

which implies that,

$$M = \frac{wx^2}{2} - P_1(x-a) = 0 \tag{3.21}$$

$$= \frac{wx^2}{2} - P_1x + P_1a = 0 \tag{3.22}$$

Therefore

$$x = -(-P_1) \pm \frac{\sqrt{P_1^2 - 2wP_1a}}{w} \qquad (3.23)$$

3.3.4 Strip Foundations

The following conditions are assumed in calculating the critical moments, according to Reynolds and Steedman [1978]:

1. Columns loading do not differ by more than 15% of the most heavily loaded column.
2. Columns spacing do not differ by more than 15% of the longest span.
3. Live load does not exceed dead load.

With the above, moments and shearing forces are calculated using the coefficients for continuous beams of three or more spans given in Table 4 of CP 110 [1972]. The critical section for positive moment is under the columns, while for negative moment it is at the center of the span. Punching shear governs the design for thickness, and it is checked as for single pads.

3.3.5 Raft Foundations

With the same assumptions as for strip foundations regarding column spacing and loading, rafts are analyzed and designed as inverted flat slabs subjected to the earth bearing pressures. Consider a raft shown in figure 3.4

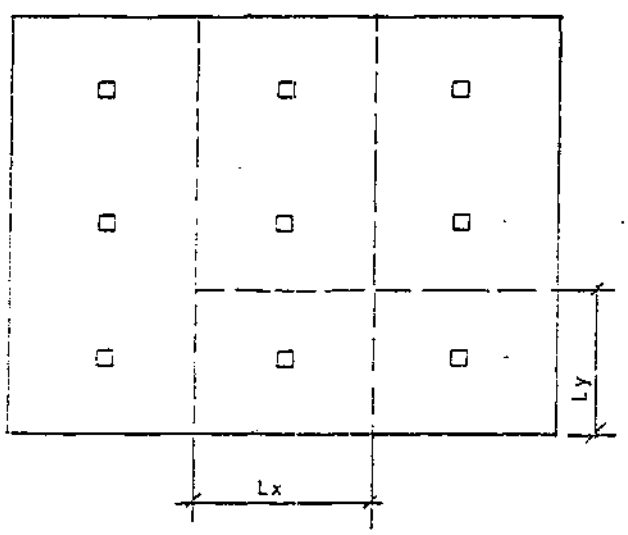


Figure 3.4 Typical raft foundation

The design moment in the x-direction is,

$$M_{ds} = \frac{1}{8}\alpha n L_y (L_x - \frac{2}{3}h_c)^2 \qquad (3.24)$$

where n = earth pressure.

α = coefficient of apportionment of moment

h_c = diameter of circular column of equivalent area

Other symbols as in figure 3.4

The design moment is apportioned between the

column and middle strips using the values of α given in table 17 of CP 110 [1972]. Punching shear governs the design for thickness, and it is calculated as for strip foundations, except that ultimate shear should be increased by 25 percent [CP 110, 1972].

With these principles for the design of reinforced concrete shallow foundations in mind, the computer program FOUND was developed so as to reduce to the barest minimum, time needed to design any or all of these foundations.

CHAPTER 4

DESIGN ALGORITHM AND PROGRAM DEVELOPMENT

4.1 Introduction

FOUND is a simple engineering structural analysis and design program that has been developed to computerize traditional methods of analysis and design of shallow building foundations that are subjected to axial loads.

4.2 Program Logic

The general idea of the program is to design spread footings for the columns of a given building and print results that would be used to prepare the structural working drawings of the building foundation.

The main input required is information on the loading and relative location of each column. Loading information required is axial loads (dead and imposed), while information on column location consists of the coordinates of the column lower left corner, and its dimensions (considering column section on plan). Other information required is soil bearing capacity and strength indices of the main construction materials (steel and concrete).

The program first assumes a single footing for each column and with the information above it determines the plan dimensions and relative location

(i.e. coordinates) of the footings. These are then stored in arrays.

The user then chooses actual footing types interactively with the computer. Having decided the footing types, the computer then designs them.

4.3 Program FOUND

4.3.1 Running the program

To run the program simply type FOUND then RETURN key.

4.3.2 Viewing results

Results are stored in a file named OUT. To view the results, type EDIT OUT, and to have a hard copy print the file OUT.

4.3.3 Program FOUND

The program consists of a main program called FOUND and nine subroutines viz.:

i) CHOOSE

ii) CHOOSE2

iii) CHOOSE3

iv) DESIGN

v) DESIN1

vi) DESIN2

vii) DESIN3

viii) DESIN4

ix) REED

Variables are mainly passed between the

subroutines through COMMON statements, except for a few cases where they are passed as arguments.

4.3.4 Arrays

The program accomplishes its task by storing and retrieving data from the following main arrays:

i) AR:- This is a 99 x 10 array. It stores the properties of the footing corresponding to the array row number. Columns 1 to 10 respectively represent:

1 Footing thickness in mm

2 Footing length in x-direction in m

3 Footing length in y-direction in m

4 X coordinate of left edge of footing

5 X coordinate of right edge of footing

6 Y coordinate of lower edge of footing

7 Y coordinate of upper edge of footing

8 Volume of concrete required for the footing in m^3

9 Weight of reinforcement in the footing in kg

10 Ratio of reinforcement weight to concrete volume in kg/m^3

ii) CLPR: - This is a 99 x 6 array. Input data on column locations and loads are stored here. The row number corresponds to the column number, while the array columns 1 to 6 respectively represent:

1 X coordinate of column left edge

2 Y coordinate of column lower edge

3 Length of column in the x-direction in m

4 Length of column in the y-direction in m

5 Column dead load in kN

6 Column live load in kN

iii) IST: - This is a one-dimensional array with a maximum number of 99 elements. It stores information on which columns share the same footing for the purpose of design.

Each element of the array represents information corresponding to the column whose number is the same as the element subscript number. The elements are assigned a number from 1 to 99 depending on the footing on which the corresponding column is located. The values are assigned by initially assuming all the columns to be on a single pad footing. Thus each element will initially be assigned the same number as the column corresponding to it, that is, IST (1) will be assigned 1, IST (2) will be 2, IST (3) will be 3 and so on. After the actual type of footing for each column is decided, the value of the IST elements will be changed to reflect this. Whenever more than one column are on the same footing, the numerically lowest column will retain its assumed IST value, while the other columns' will be changed to this value. For example if columns 3, 5, 8 and 11 are on the same footing IST (3) will remain 3, while IST (5), IST (8) and IST (11) will all be changed to 3.

iv) NTYPE: - This is a one-dimensional array of 99 elements.

It stores information on the type of footing a column is on. There are four types of footing, represented by numbers; 1 represents a single footing, 2 a combined footing, 3 a strip footing and 4 a raft. Each element will have a value equal to one of these numbers, depending on what type of footing the column numerically equal to the element subscript number is.

v) IAR: - This is a 99 x 16 array. It contains the following information on bottom reinforcement in the x direction, bottom reinforcement in the y direction, and the main top bars.

1 Area of bottom reinforcement in x direction

2 Number of bars in x direction

3 Size of the bars

4 Spacing of the bars

5 Steel ratio

6 Area of bottom reinforcement in y direction

7 Number of bars in y direction

8 Size of the bars

9 Spacing of the bars

10 Steel ratio

11 Area of top reinforcement

12 Number of top bars

13 Size of the bars

14 Spacing of the bars

15 Steel ratio

16 Number of columns on footing NFTNG

4.3.5 Subroutines

The program has been structured in such a way that the main distinct tasks are performed by subroutines. The main program coordinates execution by invoking the appropriate subroutine at the appropriate time. The main tasks of the program and the subroutines responsible for them are:

i) Reading Input Data: -

The subroutine that does this is REED. In addition to this, it also initializes the arrays. As the main program calls it, it passes the argument DAT to it. The value of the argument determines whether input would be read interactively or through a data file. If DAT equals 'S' data would be read from the screen, and the user would be asked to enter number of columns and the following for each column (default values are given in brackets):

1 Coordinates of lower left corner (25,25)

2 Dimensions in x and y directions (0.23,0.23)

3 Characteristic dead and live loads (58,24)

4 Size of dowel bars (16)

5 Characteristic cube strength (25)

6 Characteristic steel strength (410)

7 Soil allowable bearing capacity (150)

The default value for number of columns is 1. If

DAT equals 'F' data would be read from a data file. Four read statements do this. The first read statement reads in the number of columns. The second and third are executed for each column, and they read in the following:

2nd: - items 1 to 3 above

3rd: - items 4 to 6 above

The fourth reads in the allowable bearing capacity.

ii) Choosing Foundation Types

Choosing foundation types is done by a group of 3 subroutines:

1 CHOOSE

2 CHOOSE2

3 CHOOSE3

It is done interactively with the computer by answering the questions asked by the computer.

1 CHOOSES: This is the main subroutine that coordinates the choice of foundation. It starts by giving instructions on how to specify foundation type for a given column. By carrying out the instructions the subroutine would know which other subroutine to invoke in order to fix the columns on their foundations.

2 CHOOSE2: This subroutine asks the user to enter the second column of the pair on a combined footing.

The first is given to it as its argument by CHOOSE. The subroutine then changes the IST value of the second column to reflect its footing.

3 CHOOSE3: This subroutine asks for all the columns on the same strip or raft as the column given to it as argument by CHOOSE. It then changes their IST values to reflect this.

iii) Analyzing and Designing the Foundation

At this stage the NTYPE and IST values for each column would have been determined, i.e. the type of footing and columns that share the same footing would have been determined. The next stage is to design the footings, and this is done by a group of five subroutines:

1. DESIGN: This is the subroutine that coordinates the design process. It is called for each column by the main program except columns whose IST values are not equal to the column numbers, i.e. columns that had been joined with numerically lower ones on a common footing. Depending on the NTYPE value of the argument column (the column for which it was called), the subroutine calls one of the following four subroutines:

2. DESIN1: This subroutine designs the footing for columns on a single pad. After the design it stores the output in arrays that are transferred to the main program through a COMMON statement for printing.

3. DESIN2: For a column that DESIGN finds its NTYPE value to be equal to 2, implying that the column is on a combined footing, it calls DESIN2. This subroutine then searches through the IST values of all the other columns until it finds the second column on the combined footing. It then designs the footing and stores the output in arrays that are passed to the main program for printing.

4. DESIN3: This is called by DESIGN for a column whose NTYPE value equals 3, implying a strip footing. It then searches through the IST values of all the other columns until it identifies all those on this strip. The strip is then designed by it and the output stored in arrays.

5. DESIN4: This is called by DESIGN for a column whose NTYPE value equals 4, implying a raft footing. It then searches through the IST values of all the other columns until it identifies all those on this raft. The raft is then designed by it and the output stored in arrays.

4.4 Parameter Definition of Program FOUND

Parameter listing and definition is given in appendix I.

4.5 Program Listing

The flowchart of the main program is given in appendix II, while the listing of the source code of

Program FOUND is given in appendix III.

4.6 Typical Design Examples Of Shallow Foundations For Structures

The computer program FOUND was used to design the examples for a pad, combined and strip footings that are given in chapter 10 of reference 8 (Mosley & Bungey). The output from the computer is compared with the manual calculations. To check the program for a raft footing, it was used on the data for the design of the foundation of a 7-storey building. The manual design of the footings is as follows:

4.6.1 Single footing

Design parameters:

Column size: 400 mm x 400 mm

Loading: G_k = 1000 kN

Q_k = 350 kN

Materials: F_y = 425 kN/mm^2

F_{cu} = 25 N/mm^2

Dowels: 16 mm

Allowable bearing capacity = 200 kN/m^2

Manual Calculations

1 Base area: Assume a footing weight, say 15% of dead load

= 150 kN

Design axial load = 1150 + 350 = 1500 kN

Required base area = 1500/200 = 7.5 m^2

$$= \quad 2.74 \times 2.74$$

Provide a base $= \quad 2.8 \times 2.8\ m^2$

2. Footing thickness:

Compression anchorage length

$$= 24 \times 16$$

$$= 384\ mm$$

Allow 50 mm cover for two layers of reinforcement

$$= 484\ mm$$

Assume overall footing thickness of 600 mm

3. Net upward pressure:

At ultimate state design load

$$= 1.4 \times 1150 + 1.6 \times 350$$

$$= \quad 2170\ kN$$

Earth pressure $= \quad \frac{2170}{2.8^2}$

$$= \quad 277\ kN/m^2$$

Therefore net upward pressure $= 277 - 0.6 \times 24 \times 1.4$

$$= \quad 257\ kN/m^2$$

4. Bending reinforcement:

At the column face

$$M = \quad (257 \times 2.8 \times 1.2) \times \frac{1.2}{2}$$

$$= \quad 518\ kN\text{-}m$$

$$M_u = \quad 0.15 \times 25 \times 2800 \times 520^2 \times 10^6$$

$$= \quad 2839\ kN\text{-}m$$

$$M < M_u$$

$$A_s = \frac{518 \times 10^6}{(0.87 \times 425 \times (0.95 \times 520)}$$

= 2836 mm^2

Provide 10 T 20 bars at 300 mm

A_s Provided = 3140 mm^2

100 A_s /bd = 0.22 > 0.15

Maximum spacing = 185 x 2 = 370
(From table 4.1 below)

Table 4.1 - Maximum spacing of reinforcement for No redistribution of moments

FY (N/mm^2)	Maximum spacing s (mm)
250	300
425	180
400	165

Notes

1 If thickness of footing, h, $\leq$ 200 mm or h $\leq$ 250 mm and $F_y \leq 425$, then S > 3d

2 If steel ratio < 0.5, then S = 2 x the value in table 4.1

3 If 0.5 < steel ratio < 1.0 S = spacing in table 4.1 divided by steel ratio

5. Punching Shear:

Effective depth = 600 - 80 = 520 mm

Critical perimeter = 4x 400 + 3_π x 600

= 7250 mm

Area within perimeter

$=(400+3h)^2 - (4_\pi)(1.5x600)^2$

= 4.1 x 10^6 mm^2

Punching shear force = $257(2.8^2 - 4.1)$

= 961 kN

Punching shear stress = $\frac{961 \times 10^3}{(7250 \times 520)}$

= 0.26 N/mm^2

Ultimate shear stress = 0.35 N/mm^2 >0.22

(From table 4.2 below)

Table 4.2 Ultimate shear stress values (N/mm^2)

Steel ratio	Concrete grade			
	20	25	30	40 or more
0.25	0.35	0.35	0.35	0.35
0.50	0.45	0.50	0.55	0.55
1.00	0.60	0.65	0.70	0.75
2.00	0.80	0.85	0.90	0.95
3.00	0.85	0.90	0.95	1.00

6. Local bond at critical section for bending

$V = 257 \times 2.8 \times 1.2 = 864$ kN

$$F_{bs} = \frac{864 \times 10^3}{(10 \times 62.8 \times 520)}$$

$= 2.65\ N/mm^2$

Ultimate bond stress = $3.0\ N/mm^2 > 2.65$

(From table 4.3 below)

Table 4.3 Ultimate local bond stress (N/mm^2)

Bar type	Concrete grade			
	20	25	30	40 or more
Plain bars	1.7	2.0	2.2	2.7
Deformed bars, type 1	2.1	2.5	2.8	3.4
Deformed bars, type 2	2.5	3.0	3.4	4.1

7. Shear stress at critical section

Shear force, $V = 257 \times 2.8 \times 0.42$

$= 302$ kN

Shear stress, $v = \frac{302 \times 10^3}{2800 \times 520}$

$= 0.21\ N/mm^2 < 0.35$

Computer Solution

Input file for single footing

```
1
5 5 .400 .400 1000 350
16
200
25 425
```

Results of FOUND for single footing

FT (mm)	XL (m)	YL (m)	VOL (m^3)	WT (kg)	WT/VOL (kg/m^3)
654.77	2.63	2.63	4.51	132.80	29.43

REINFORCEMENTS

LONGITUDINAL						TRANSVERSE		
BOTTOM			TOP			BOTTOM		
NO	SIZE (mm)	SPA (mm)	NO	SIZE (mm)	SPA (mm)	NO	SIZE (mm)	SPA (mm)
16	16	175	0	0	0	16	16	175

4.6.2 Combined footing

Design parameters:

Column size: 1 300 x 300

2 400 x 400

Loading: 1 Gk = 1000 kN

Qk = 200 kN

2 Gk = 1400 kN

Qk = 300 kN

Materials: Fy = 410 N/mm^2

F_{cu}= 30 N/mm^2

Dowels: 25 mm

Allowable bearing capacity: 300 kN/m^2

Manual Calculations

1. Base area:

Self weight of footing - allow say 10% of dead load

= 240; say 250 kN

Therefore total load = 250 + 1000 +200

+ 1400 + 300

= 3150 kN

Base area required = $\frac{3150}{300}$

= $10.5 m^2$

Provide a base = 4.6 m x 2.3 m

Area = $10.58 m^2$

2. Resultant of column loads at centroid of base:

Taking moments about 400 mm column gives

$$\bar{x} = \frac{1200 \times 3}{1200+1700} = 1.24$$

Therefore, base is centred 1.24 m from centre line of 400 mm column

3. Bearing pressure:

Total ultimate load = 1.4 (1000 + 1400+250)

+ 1.6 (200 + 300)

= 4510 kN

Therefore earth pressure = $\frac{4510}{10.58}$

= $426 kN/m^2$

Net upward pressure = 426 - 1.4 x 0.8 x 24

= $400 kN/m^2$

4. Base thickness: assume bar diameter of say 25mm

Therefore compression anchorage length

= 21x25

= 525 mm

Provide a base thickness of say 800 mm to allow for cover and two layers of reinforcement

5. Moments and shear forces are shown in figure 4.1 below

6. Shear stress

Effective depth = 740mm

Therefore shear force

$$V = 1509 - 400 \times 2.3(1.5 \times 0.74 + 0.2)$$

$$= 304 \text{ kN}$$

Shear stress, $$v = \frac{304 \times 10^3}{2300 \times 740}$$

$$= 0.18 \text{ N/mm}^2$$

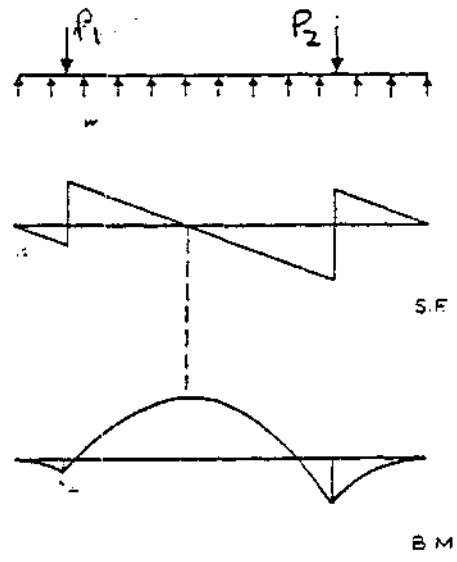


Figure 4.1 Bending moment and shear force diagram

7. Bending reinforcement

At mid-span between columns

$$As = \frac{717 \times 10^6}{0.87 \times 425 \times 0.95 \times 740}$$

$$= 2759 \text{ mm}^2$$

Provide 9T20 @ 270 mm

Area provided = 2830 mm^2

Local bond stress = $\frac{1151 \times 10^3}{565 \times 740}$

= 2.75 N/mm^2

Ultimate bond stress = 3.4 N/mm^2

> 2.75

(from Table 4.3)

At the face of the 400 mm column

$M = 400 \times 2.3 \times \frac{(1.06 - 0.2)^2}{2}$

= 340 kN-m

$A_s = \frac{340 \times 10^6}{0.87 \times 425 \times 0.95 \times 740}$

= 1309 mm^2

$A_{smin} = \frac{0.15 \times 2300 \times 740}{100}$

= 2553 mm^2

Provide 9T20 @ 270 mm

A_s = 2830 mm^2

Local bond stress = $\frac{400 \times 2.3 \times 0.86 \times 10^3}{9 \times 62.8 \times 740}$

= 1.89 N/mm^2 < 3.36

8. Transverse bending

$M = 400 \times \frac{1.15^2}{2}$

= 265 kN-m/m

$A_s = \frac{265 \times 10^6}{0.87 \times 460 \times 0.95 \times 720}$

= 969 mm^2/m

A_{smin} = 1080 mm^2/m

Provide T16 @ 180 mm

Area = 1117 m^2/m

Local bond stress = $\frac{400 \times 1.15 \times 10^3}{279 \times 720}$

= 2.29 N/mm^2

Computer Solution

Input file for combined footing

```
2
8 7 .300 .300 1000 200
25
10.95 6.95 .400 .400 1400 300
25
200
30 425
```

Results of FOUND for combined footing

FT (mm)	XL (m)	YL (m)	VOL (m^3)	WT (kg)	WT/VOL (kg/m^3)
756.00	4.53	2.27	7.77	269.30	34.64

REINFORCEMENTS

LONGITUDINAL						TRANSVERS		
BOTTOM			TOP			BOTTOM		
NO	SIZE (mm)	SPA (mm)	NO	SIZE (mm)	SPA (mm)	NO	SIZE (mm)	SPA (mm)
12	16	187	9	20	250	23	16	196

4.6.3 Strip footing

Design parameters:

Column size: 400 mm x 400 mm

Column spacing: 3.5 m centers

Loading: G_k = 1000kN

Q_k = 350kN

Materials: F_y = 425 kN/mm^2

F_{cu} = 25 N/mm^2

Dowels: 16 mm

Allowable bearing capacity = 200 kN/m^2

Manual Calculations

Try a thickness of footing = 700 mm

Effective depth = 620 mm

Assume footing self-weight = 40 kN/m

Width of footing required = $\frac{1000 + 350 + (40 \times 3.5)}{200 \times 3.5}$

= 2.13 m

Provide a strip footing = 2.2 m wide

Bearing pressure at ultimate state =

$$\frac{1.4(1000+40\times3.5)+1.6\times350}{2.2 \times 3.5}$$

= 280 kN/m^2

Net upward pressure = 280 -1.4x0.7x24

= 257 kN/m^2

Longitudinal Reinforcement:

Using the moment and shear coefficients for an equal span continuous beam from figure 4.2 below:

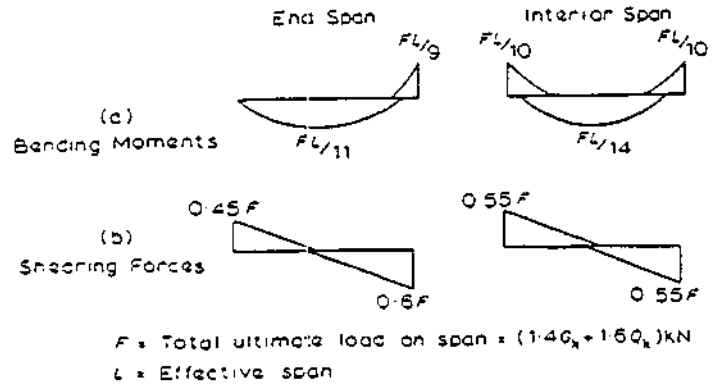


Figure 4.2 Moment and shear coefficients for an equal span continuos beam

For an interior span

Moment at the columns = $\frac{257 \times 2.2\ 3.5^2}{10}$

= 693 kN-m

Therefore A_s = $\frac{693 \times 10^6}{0.87 \times 425 \times 0.95 \times 620}$

= 3183 mm^2

Minimum A_s = 2050 mm^2

Provide 11 T 20 at 200 mm centres as bottom steel

Area = 3454 mm^2

In the span M = $\frac{257 \times 2.2 \times 3.5^2}{14}$

= 495 kN-m

Therefore A_s = $\frac{495 \times 10^6}{0.87 \times 425 \times 0.95 \times 620}$

= 2273 mm^2

Provide 8 T 20 bars at 280 mm centres as top steel

Area = 2512 mm^2

Transverse reinforcement

M = $257 \times \frac{1.1^2}{2}$

= 156 kN m/m

A_s = $\frac{156 \times 10^6}{0.87 \times 425 \times 0.95 \times 620}$

= 684 mm^2/m

Minimum A_s = $\frac{0.15 \times 1000 \times 600}{100}$

= 900 mm^2

Provide T16 bars at 220 mm centres, A_s = 914 mm^2

Shear:

1.5d from column face

$V = 257 \times 2.2 \ (3.5 \times 0.55 - 1.5 \times 0.62)$

$= 450$ kN

Shear stress $v = \frac{450 \times 10^3}{2200 \times 620}$

$= 0.33$ N/mm^2

Allowable shear= 0.35

(from table 4.2)

Computer solution

Input file for strip footing

```
4
3 9 .400 .400 1000 350
16
3 12.5 .400 .400 1000 350
16
3 16 .400 .400 1000 350
16
3 19.5 .400 .400 1000 350
16
200
25 425
```

Results of FOUND for Strip footing

FT (mm)	XL (m)	YL (m)	VOL (m^3)	WT (kg)	WT/VOL (kg/m^3)
633.21	12.50	2.09	16.52	589.82	35.70

REINFORCEMENTS

LONGITUDINAL						TRANSVERSE		
BOTTOM			TOP			BOTTOM		
NO	SIZE (mm)	SPA (mm)	NO	SIZE (mm)	SPA (mm)	NO	SIZE (mm)	SPA (mm)
11	20	200	12	16	175	7	12	150

4.6.4 Raft Foundation

Design parameters:

Total service loads : Dead = 3056 kN

Live = 917 kN

Column sizes = 750 x 750 mm^2

Column spacing = 7.2 m

Allowable bearing capacity = 200 kN/m^2

Manual Calculations

Try 700 mm deep raft

Earth pressure at service loads =

$$\frac{(7.2^2 \times 0.7 \times 24) + 3056 + 917}{7.2^2}$$

$$= 93.44 \text{ kN/m}^2$$

$$M_{ds} = \frac{1}{8}\alpha n L_Y (L_X - \frac{2}{3} h_c)^2$$

$$h_c = 0.4$$

Earth pressure at ultimate load =

$$1.4(3056+870.9)+1.6(917)$$

$$= 134.35 \text{ kN/m}^2$$

Net upward pressure, n =

$134.35 - 24 \times 1.4 \times 0.7$

$= 110.83 \text{ kN/m}^2$

Column strip moment :

Positive, M_c^+ =

$0.125 \times 0.46 \times 110.83 \times 7.2 \times 6.94^2$

$= 2208.85 \text{ kN-m}$

$$A_s = \frac{2208 \times 10^6}{0.95 \times 0.87 \times 410 \times 640}$$

$= 10181 \text{ mm}^2$

$$\frac{100A_s}{bd} = 0.44$$

Ultimate shear = 1.25×0.4

(from table 4.2)

$= 0.5 \text{ N/mm}^2$

Check punching shear:

$$P_{cr} = 4 \times 750 + 3\pi \times 700 = 9600 \text{ mm}$$

$$A_{cr} = (750 + 3h)^2 - (4 - \pi)(1.5h)^2 = 7.17 \times 10^6 \text{ mm}^2$$

Punching shear force, V

$= 110.83 \ (7.2^2 - 7.17) = 4950.78 \text{ kN}$

Punching shear stress, v

$$= \frac{4950.78 \times 10^3}{9600 \times 640}$$

$= 0.81 \text{ N/mm}^2 > 0.5$

Try a 900 mm deep raft

$P_{cr} = 4 \times 750 + 3\pi(900)$

$= 11478 \text{ mm}$

$$A_{cr} = (750 + 3x900)^2 - 0.86(1350)^2$$

$$= 10.34 \times 10^6 \text{ mm}$$

$$PSF = 110.83(51.8 - 10.38)$$

$$= 4595.01 \text{ kN}$$

$$PSS = \frac{4.59 \times 10^6}{11478 \times 840}$$

$$= 0.48 \text{ N/mm}^2$$

$$A_s = 10181 \times \frac{640}{840}$$

$$= 7756.95 \text{ mm}^2$$

$$\frac{100A_s}{bd} = 0.27$$

Allowable shear = 1.25 x 0.35

= 0.44 N/mm²

< 0.48

Increase $\frac{100A_s}{bd}$ to 0.5

Allowable shear = 1.25 x 0.5

= 0.63 N/mm²

> 0.48

Provide Y32 @ 200 mm centers under the columns

A_s = 4020 mm²/m

Other reinforcements:

Column strip -ve moment, M_c^- = 2208 x 0.22/0.46

= 1056 kN-m

$$A_s = \frac{1056 \times 10^6}{0.95x0.87x410x840}$$

= 3709.86 mm²

Min A_s = 0.0015 x 3600 x 840

= 4536 mm^2

Provide Y20 @ 250 mm centers, A_s = 1260 mm^2/m

Middle strip

Negative moment, M_s^- = 2208.85 x $\frac{0.16}{0.46}$

= 768.3 kN-m

Provide Y20 @ 250 mm centers, A_s = 1260 mm^2/m

Computer solution

Input file for raft

```
9
4.5 9.5 .750 .750 3056 917
25
11.7 9.5 .750 .750 3056 917
25
18.9 9.5 .750 .750 3056 917
25
4.5 16.7 .750 .750 3056 917
25
11.7 16.7 .750 .750 3056 917
25
18.9 16.7 .750 .750 3056 917
25
4.5 23.9 .750 .750 3056 917
25
11.7 23.9 .750 .750 3056 917
25
18.9 23.9 .750 .750 3056 917
25
250
25 425
```

Results of FOUND for raft

FT (mm)	XL (m)	YL (m)	VOL (m3)	WT (kg)	WT/VOL (kg/m3)
756.00	17.15	17.15	222.36	5535.55	24.89

RAFT REINFORCEMENTS

REINFORCEMENT IN X-DIRECTION

	COLUMN STRIP						MIDDLE STRIP					
PANEL	BOTTOM			TOP			BOTTOM			TOP		
	NO	SIZE mm	SPA mm	NO	SIZE mm	SPA mm	NO	SIZE mm	SPA mm	NO	SIZE mm	SPA mm
1	32	32	166	7	16	166	5	16	200	5	16	200
2	32	32	166	7	16	166	5	16	200	5	16	200
3	32	32	166	7	16	166	5	16	200	5	16	200
4	32	32	166	7	16	166	5	16	200	5	16	200

REINFORCEMENT IN Y-DIRECTION

	COLUMN STRIP						MIDDLE STRIP					
PANEL	BOTTOM			TOP			BOTTOM			TOP		
	NO	SIZE mm	SPA mm	NO	SIZE mm	SPA mm	NO	SIZE mm	SPA mm	NO	SIZE mm	SPA mm
1	32	32	166	7	16	166	5	16	200	5	16	200
2	32	32	166	7	16	166	5	16	200	5	16	200
3	32	32	166	7	16	166	5	16	200	5	16	200
4	32	32	166	7	16	166	5	16	200	5	16	200

4.7 Comparison of manual results and results from FOUND

A comparison of results obtained from the computer program FOUND and manual calculations are shown in table 4.1 below.

Table 4.1 - comparison of manual and computer calculations

FOUNDATION	FOOTING AREA (mm^2)			FOOTING THICKNESS (mm)			BOTTOM REINF. (mm^2)			TOP REINF. (mm^2)			TRANS. REINF. (mm^2)		
	man	com	% diff	man.	com	% diff	man	com	% diff	man	com	% dif	man	com	% dif
SINGLE	7.84	6.92	+13.29	600.00	654.77	-8.36	3140	321 7	-2.39	0.00	0.00	0.00	3140	3217	-2.4
COMBINED	10.58	10.28	+2.92	800.00	756.00	+5.82	2830	2413	+17.28	2830	3142	-9.93	1117	1005	+11
STRIP	7.70	7.32	+6.14	700.00	633.21	+10.55	1570	1570	0.00	1141.8	1150.	+7.11	915	754	+21.4
							RAFT REINFORCEMENT								
RAFT							Column strip						middle strip		
							Top			bottom			top		
							man	com	% diff	man.	com.	% dif	man	com	% dif
				900.00	756.00	19.05	1260	1210	+4.0	4020	4843	+17.0	1260	1010	+24.0

% difference in relation to computer solution
+ Implies manual result is higher
- Implies computer result is higher
man. - manual calculation
com. - computer calculation

4.8 Discussion of results

Generally, FOUND gives a more economical section in terms of volume of concrete. The amount of main steel reinforcement is expectedly higher because of the shallower depth of footing. FOUND therefore gives an optimized section. A detailed discussion on each of the four types of footings is given below.

4.8.1 Single footing

(i) Footing Area - The actual footing area from manual solution (MS) is 7.5 m^2 against 6.92 m^2 from computer solution (CS). The large difference between the CS and the MS is because the actual area from manual computation was rounded up to 7.84 m^2 in order to have a convenient footing dimension of 2.8 x 2.8 m. But even considering the actual calculated area from MS, it is still higher than that from CS by 8.5%. This is because MS assumes a footing weight of 150 kN, while CS determined the actual footing weight and used it.

(ii) Footing Thickness - The footing thickness from MS is much higher because a generous assumption was made initially, but when the actual shear stress turned out much lower than the allowable, no adjustment was made. On the other hand, with CS a small footing thickness was started with, then small increments were made until the allowable shear stress was just higher than the actual.

(iii)Reinforcement - is slightly higher from CS than from MS because a smaller effective depth occurs under CS.

4.8.2 Combined footing

(i) Footing Area - The area from MS is just slightly higher than that from CS because the assumed footing weight under MS is also fairly close to the actual weight.

(ii) Footing thickness - The thickness from MS is about 6% higher than from CS for the same reason as in single footing.

(iii) Reinforcement - The area of bottom reinforcement from MS is much higher than from CS because the actual area from MS had to be increased in order to satisfy minimum reinforcement requirement, which has in turn been affected by the higher footing thickness of MS. Conversely, top reinforcement from MS is lower because it's actual value is just slightly lower than minimum requirement. The area from CS is higher than even the minimum requirement because of the lower footing thickness, which implies lower effective depth.

4.8.3 Strip footing

(i) Footing Area - The footing area from MS is higher by 6% because the footing width, which is the

only variable that can be different, has been rounded to the convenient figure of 2.2 m.

(ii) Footing thickness - The footing thickness from MS is higher for the same reasons as in single footing

(iii)Reinforcement - The bottom reinforcement is exactly equal for the two solutions, even though the CS has a smaller effective depth. It should however be noted that the area provided under MS is higher than the actual area required by about 9%. Thus, strictly speaking, the area from CS is lower than that from MS, which as expected
(because the assumption of footing thickness in MS implied a larger bending moment on the footing).

4.8.4 Raft footing

(i) Footing area - is the same for both solutions since what is considered is the area of a panel, which is fixed.

(ii) Footing thickness - Footing thickness from MS is higher by about 11%. This is because the steel ratio from CS is higher by about 44% and expectedly this will lower the required depth of footing.

(iii)Reinforcement - The bottom reinforcement from CS is expectedly higher due to the smaller footing depth. Other reinforcements from CS are however lower because those from the MS had to be increased to

satisfy minimum requirements, which have been made higher by the footing thickness.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

The design of reinforced concrete shallow foundations for structures has been presented with the aid of a computer in FORTRAN 77 language. Suggestions and recommendations made by various scholars and institutions for the design of reinforced concrete sections have been considered and incorporated in the computer program FOUND developed.

The results obtained from the computer program turn out to be optimized versions of the manually computed results. Also, the routine of carrying out reinforced concrete foundation designs manually, which may result in some mathematical errors, is eliminated when FOUND is used. FOUND is therefore recommended for the design of reinforced concrete shallow foundations in simple grounds. Results obtained should however be rounded to up to practicable values.

5.1 Recommendation

A similar computer program should be developed for reinforced concrete foundations to incorporate the following:

- Foundations subjected to bi-axial bending
- Columns of any shape
- Use of polar coordinates to locate positions

- Design deep foundations where shallow ones are not possible solutions

REFERENCES

Abejide, O. S. (1994) **Computer-aided Design of Continuous Pre-stressed Concrete Beams.** MSc. Thesis, submitted to the postgraduate School, Ahmadu Bello University. November, 1994

Astill, A. W and Martin, L. H. (1981) **Elementary Structural Design In Concrete To CP 110.** Butterworth and Company Ltd. London.

Bensasson, S et al. (1980) **Computer Applications for detailed Design Calculations of Reinforce Concrete Structures.** Structural Engineer, Vol. 58A, No. 6. June, 1980 pp 195 - 201.

Bhatti, M. A; Pister, K. S and Polak, E.(1981) **Package for Optimization-Based Interactive CAD.** Journal of the Structural Division, Proceedings of ASCE, Vol. 107, No. ST11. November, 1981. pp 2271 - 2284.

BS 8110 (1985) **The Structural Use of Concrete: Parts 1 -3**. Her Majesty's Stationery Office London.

CP110 (1972) **The Structural Use of Concrete: Part 1; Design and Workmanship.** Her Majesty's Stationery Office London

Falk, H.(1985) **Microcomputer Software for Concrete Structural Design**. Concrete International. June, 1985. pp 4 - 56.

Mosley, W. H and Bungey, J. H.(1982) **Reinforced Concrete Design**. Macmillan Press Limited London.

Reynolds, C.E and Steedman, J.C (1981) **Reinforced Concrete Designers Hand Book.** Cement and Concrete Association London.

Rooney, M. F and Smith, S. E. (1982) **Artificial Intelligence In Engineering Design.** Computer and Structures. No. 16. pp 279 - 288.

Sandewall, E.(1978) **A Survey of Artificial Intelligence With Respect to Computer-aided Design in AI and Pattern Recognition in CAD**. (Edited by J. C. Latombe). North-Holland, Amsterdam. pp 19 - 34.

Winston, P. H.(1984) **Artificial Intelligence**. Addison-Wesley. Reading, M. A.

APPENDICES

Appendix I

Parameter listing and definition

Note that * specifies the width of the character.

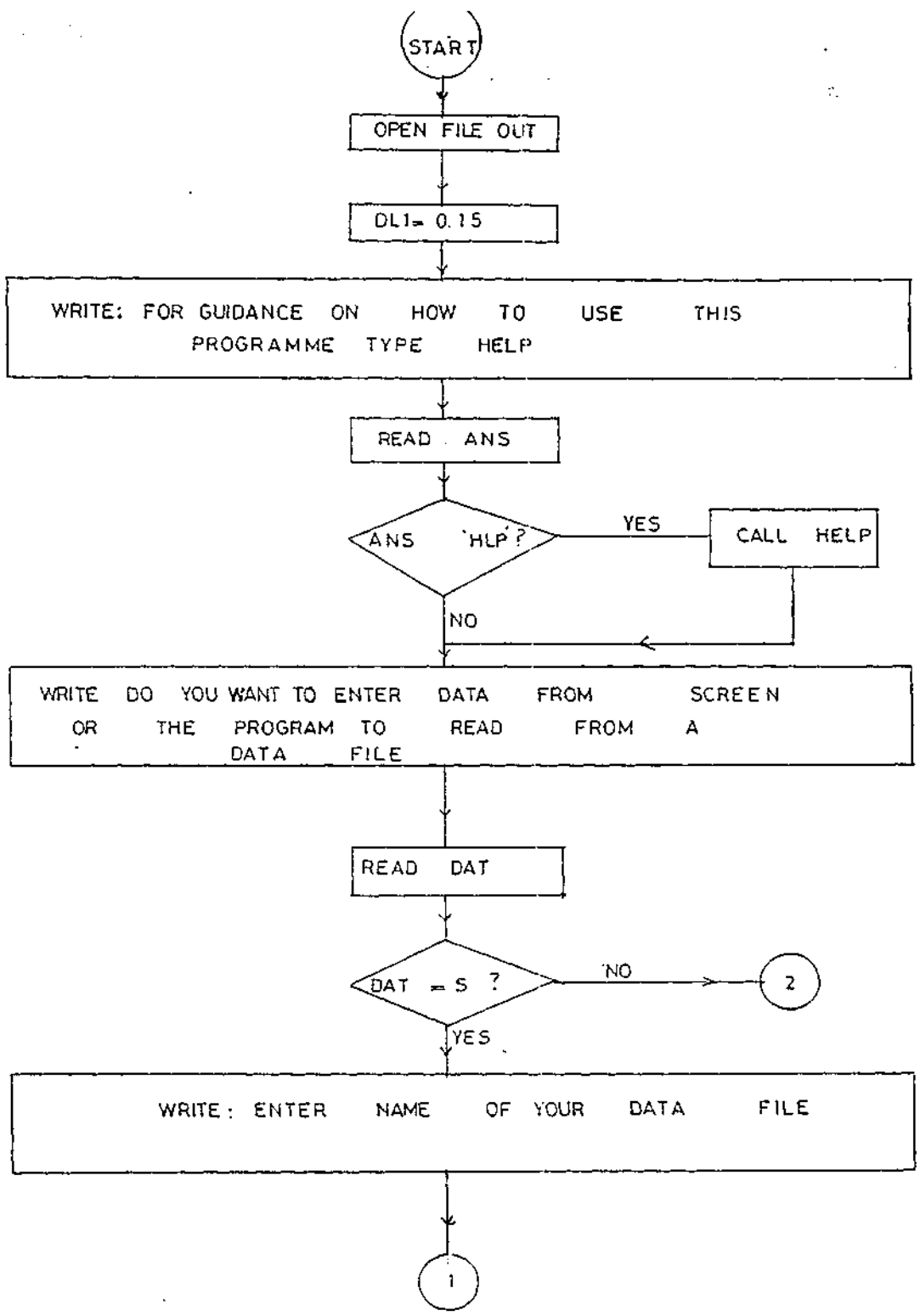
ABCAP	REAL	Allowable bearing capacity
ANS	CHAR*1	Parameter used to decide READ statements
AR	REAL	Array storing footing dimensions
BARS	REAL	Array storing bar sizes
CLPR	REAL	Array storing column dimensions and loads
DAT	CHAR*2	Parameter used in READ statements
DL1	REAL	Factor for footing weight in calculating earth pressure for single footings
I	INTEGER*4	Counter
IAR	INTEGER*4	Array storing footing reinforcements
ICOL	INTEGER*4	Counter
IPANEL	INTEGER*4	Counter
IRAF	INTEGER*4	Counter
IRX	INTEGER*4	Array storing raft reinforcements
IST	INTEGER*4	Array storing column status
M	INTEGER*4	Counter
MATRS	INTEGER*4	Array storing dowel bar size, FY, Fcu
NAME	CHAR*8	Parameter for file name
NCOLS	INTEGER*4	Counter
NFTNG	INTEGER*4	Counter
NPANEL	INTEGER*4	Array storing columns at panel corners
NPNL	INTEGER*4	Counter

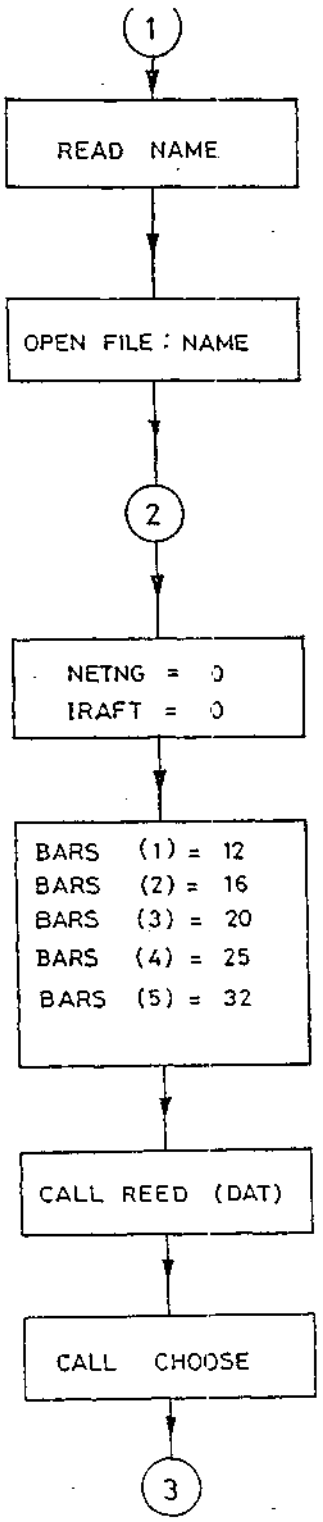
NTYPE	INTEGER*4	Array storing footing types
PMMT	REAL	Array storing panel moments
PRENF	REAL	Array storing panel reinforcements
TIPE	CHAR*8	Parameter for footing type
XMAX	REAL	Parameter for maximum x coordinate
YMAX	REAL	Parameter for maximum y coordinate
FCU	REAL	Parameter for concrete cube strength
FY	REAL	Parameter for steel strength
ABR	REAL	cross section area of bending reinforcement in x direction
ABRY	REAL	cross section area of bending Reinforcement in y direction
ASX	REAL	Area of bending reinforcement in x direction
ASXP	REAL	Area of bending reinforcement provided in x direction
ASY	REAL	Area of bending reinforcement in y direction
ASYP	REAL	Area of bending reinforcement provided in y direction
AWP	REAL	Area within perimeter
BMX	REAL	Bending moment about x axis
BMY	REAL	Bending moment about y axis
BRNX	REAL	Parameter for number of bars in x direction
BRNY	REAL	Parameter for number of bars in y direction

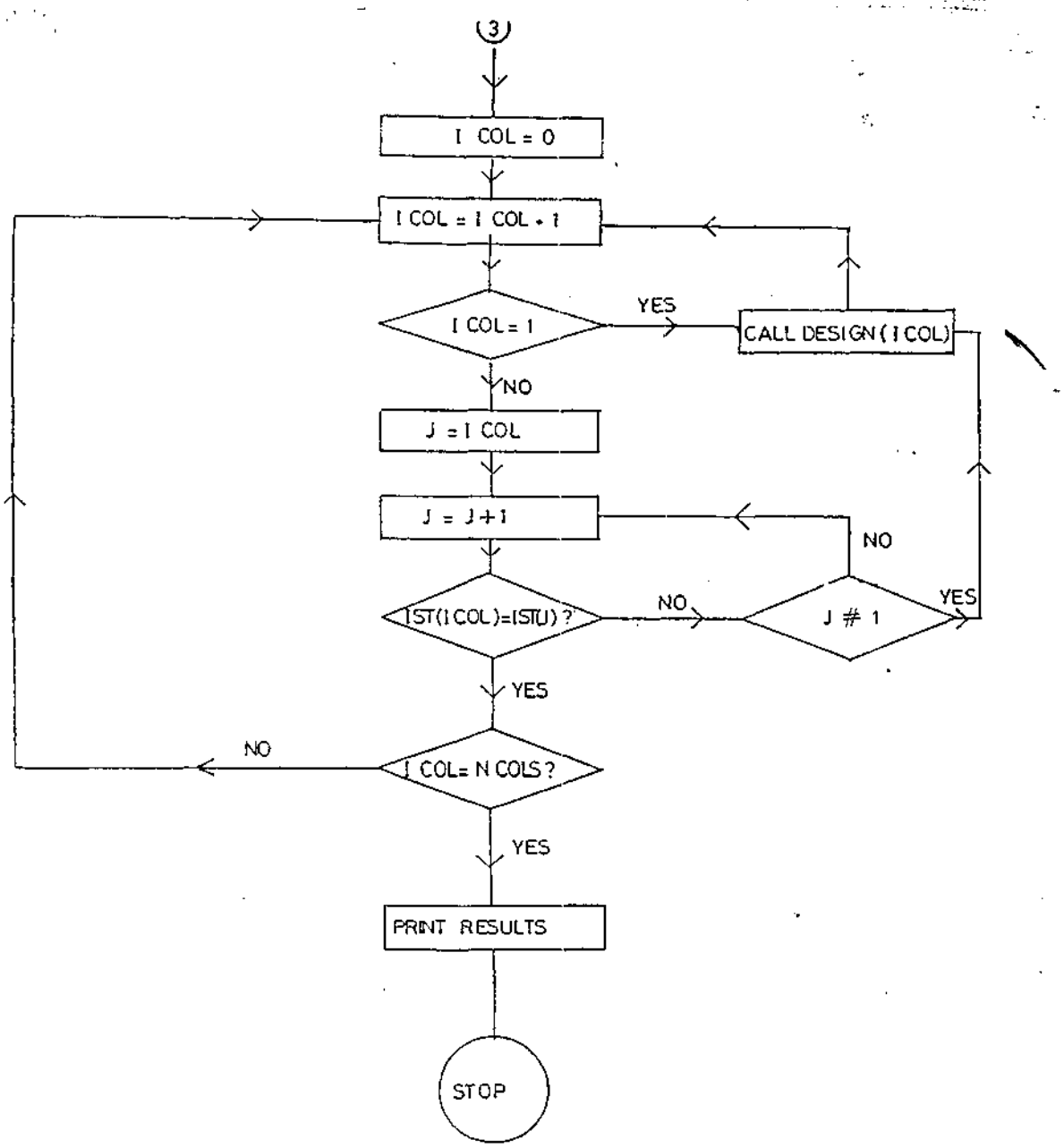
BRX	REAL	Parameter for size of reinforcement in x direction
BRY	REAL	Parameter for size of reinforcement in y direction
BSLX	REAL	Parameter for local bond stress in x direction
BSLY	REAL	Parameter for local bond stress in y direction
BT	REAL	Beta
CP	REAL	Parameter for critical perimeter
ED	REAL	Effective depth
EP	REAL	Earth pressure
FD	REAL	Coefficient for calculating shear
FT	REAL	Footing thickness
FTWT	REAL	Footing weight
PSF	REAL	Punching shear force
PSS	REAL	Punching shear stress
RK2	REAL	Coefficient for calculating shear
ROX	REAL	Steel ratio in x direction
ROY	REAL	Steel ratio in y direction
SSX	REAL	Shear stress in x direction
SSY	REAL	Shear stress in y direction
SX	REAL	Bar spacing in x direction
UBS	REAL	Ultimate bond stress
UPN	REAL	Net upward pressure
USSX	REAL	Ultimate bond stress in x direction
USSY	REAL	Ultimate bond stress in y direction

Appendix II

Flow Chart of Main Program


START
OPEN FILE OUT
DL1= 0.15
WRITE: FOR GUIDANCE ON HOW TO USE THIS PROGRAMME TYPE HELP
READ ANS
ANS 'HLP'?
YES
CALL HELP
NO
WRITE DO YOU WANT TO ENTER DATA FROM SCREEN OR THE PROGRAM TO READ FROM A DATA FILE
READ DAT
DAT = S ?
NO
2
YES
WRITE: ENTER NAME OF YOUR DATA FILE
1


1
READ NAME
OPEN FILE : NAME
2
NETNG = 0
IRAFT = 0
BARS (1) = 12
BARS (2) = 16
BARS (3) = 20
BARS (4) = 25
BARS (5) = 32
CALL REED (DAT)
CALL CHOOSE
3


3
I COL = 0
I COL = I COL + 1
I COL = 1
YES
CALL DESIGN (I COL)
NO
J = I COL
J = J+1
NO
IST(I COL)=IST(J) ?
NO
J # 1
YES
YES
I COL= N COLS ?
NO
YES
PRINT RESULTS
STOP

Appendix III

Source code of Program FOUND

```
      PROGRAM FOUND
      COMMON /A/ NCOLS,ABCAP
      COMMON /B/ XMAX,YMAX
      COMMON /D/ FOOTD(100,7),CLPR(100,12)
      COMMON /E/ IST(100),NTYPE(100),MATRS(100,3)
      COMMON /F/ AR(99,10),IAR(99,16)
      common /h/ dl1
      COMMON /P/ PMMT(10,10),PRENF(10,8)
      COMMON /Q/ BARS(5)
      COMMON /R/ IRAF(100,9),NPANEL(10),IRX(10,32)
      COMMON IRAFT
      COMMON NFTNG
      CHARACTER ANS,DAT*2,NAME*8,TIPE*8
C
C Open file for printing results
C
      OPEN(4,FILE='OUT',STATUS='NEW')
C
c dl1  makes provision for footing weight when determining
c soil pressure. It is initially taken as 15%
c
      dl1=.15
C
      WRITE(*,1)' FOR GUIDANCE ON HOW TO USE THIS PROGRAM TYPE "HLP"'
C
  1   FORMAT(A)
C
      REad(*,1)ANS
      IF(ANS .EQ. 'HLP')CALL HELP
C
      WRITE(*,1)' DO YOU WANT TO ENTER DATA FROM THE SCREEN OR THE PROGR
     1AM TO READ FROM A DATA FILE ? (S OR F)'
C
      DAT = 'F'
      REad(*,1)DAT
      IF(DAT .EQ. 'F')THEN
        WRITE(*,1)' ENTER THE NAME OF YOUR DATA FILE'
        REad(*,'(A8)')NAME
        OPEN(3,FILE =NAME)
      END IF
C
c Initialise number of footings and rafts
c
      NFTNG=0
      IRAFT=0
C
c Assign values to array BARS,which contains reinforcement
c sizes commonly usED
c
      BARS(1)=12
      BARS(2)=16
      BARS(3)=20
      BARS(4)=25
      BARS(5)=32
C
      CALL REED(DAT)
```

```
C
      CALL CHOOSE
C
c Taking columns one by one , design footing for each column
c
      ICOL = 0
      DO 20 I=1,NCOLS
         ICOL = ICOL + 1
         IF(ICOL .EQ. 1 )CALL DESIGN(ICOL)
C
c For columns other  than column one,check that they have not
c been joined with numerically lower columns on a common footing
c
         IF(ICOL .GT. 1)THEN
           DO 15 J=ICOL-1,1,-1
c
c If joined with another column do not call DESIGN
c
             IF(IST(ICOL) .EQ. IST(J))THEN
               GOTO 20
             END IF
  15       CONTINUE
           CALL DESIGN(ICOL)
         END IF
  20  CONTINUE
C
c Print results
c
      WRITE(4,'(A/)')' FTNG       PT            XL            YL           VOL

     1         WT          WT/VOL      '
      WRITE(4,'(A/)')'            (mm)          (m)           (m)          (m3)
     1       (kg)          (kg/m3)'
      N=0
      DO 21 I=1,NFTNG
        N=N+1
        WRITE(4,'(I2,6F12.2)')N,(AR(N,J),J=1,3),(AR(N,J),J=8,10)
 21   CONTINUE
C
      II=0
      DO 22 I=1,NFTNG
       IF(NTYPE(I) .EQ. 1 .OR. NTYPE(I) .EQ. 2 .OR. NTYPE(I) .EQ. 3)THEN
      II=II+1
      WRITE(4,'(//A/)')' REINFORCEMENTS'
       WRITE(4,'(A)')'                ³             LONGITUDINAL

      1³   TRANSVERS '
       WRITE(4,'(A)')'                ³                  ³

      1³'
       WRITE(4,'(A)')'                ³     BOTTOM       ³      TOP

      1³     BOTTOM'
       WRITE(4,'(A)')'                ³                  ³

      1³'
       WRITE(4,'(A)')'FTNG TYP  NCOLS³ NO   SIZE   SPA ³ NO   SIZE   SPA
      1³ NO   SIZE   SPA'
        IF(II .GT. 1)GOTO 23
        END IF
 22    CONTINUE
 23    CONTINUE
C
      IF(NTYPE(1) .NE. 4)THEN
       IF(NTYPE(1) .EQ. 1)TIPE='   SINGL'
```

```
      IF(NTYPE(1) .EQ. 2)TIPE='  COMBD'
      IF(NTYPE(1) .EQ. 3)TIPE='  STRIP'
      WRITE(4,'(A,A,I3,A,3(I3,I6,I7,A))')' 1',TIPE,IAR(1,16),'  ³',(IAR
     1 (1,J),J=2,4),' ³',(IAR(1,J),J=12,14),' ³',(IAR(1,J),J=7,9)
      END IF
C
      IF(NFTNG .GT. 1)THEN
      N=1
      DO 25 I=2,NFTNG-1
       N=N+1
       IF(IRAFT .EQ. 0)THEN
       IF(NTYPE(N) .EQ. 1)TIPE='  SINGL'
       IF(NTYPE(N) .EQ. 2)TIPE='  COMBD'
       IF(NTYPE(N) .EQ. 3)TIPE='  STRIP'
       WRITE(4,'(I2,A,I3,A,3(I3,I6,I7,A))')  N,TIPE,IAR(N,16),'  ³',(IAR
     1 (I,J),J=2,4),' ³',(IAR(I,J),J=12,14),' ³',(IAR(I,J),J=7,9)
       END IF
 25   CONTINUE
C
      K=1
      DO 27 N=2,NFTNG-1
      K=K+1
       IF(IRAFT .NE. 0)THEN
           IF(NTYPE(N-1).EQ. 1 .OR. NTYPE(N+1).EQ. 1 .OR. NTYPE(N-1).EQ.
     1     2 .OR. NTYPE(N+1).EQ. 2 .OR. NTYPE(N-1).EQ. 3 .OR. NTYPE(N+1)
     2.EQ. 3)THEN
            IF(NTYPE(N) .EQ. 1)TIPE='  SINGL'
            IF(NTYPE(N) .EQ. 2)TIPE='  COMBD'
            IF(NTYPE(N) .EQ. 3)TIPE='  STRIP'
            IF(NTYPE(N) .EQ. 4)THEN
             TIPE='  RAFT'
             WRITE(4,'(I2,A,A)')K,TIPE,'  SEE RAFT'
             GOTO 26
            END IF
       WRITE(4,'(I2,A,I3,A,3(I3,I6,I7,A))')K,TIPE,IAR(N,16),'  ³',(IAR
     1 (I,J),J=2,4),' ³',(IAR(I,J),J=12,14),' ³',(IAR(I,J),J=7,9)
           END IF
       END IF
  26   CONTINUE
  27   CONTINUE
C
      IF(NTYPE(NFTNG) .NE. 4 )THEN
       IF(NTYPE(NFTNG) .EQ. 1)TIPE='  SINGL'
       IF(NTYPE(NFTNG) .EQ. 2)TIPE='  COMBD'
       IF(NTYPE(NFTNG) .EQ. 3)TIPE='  STRIP'
       WRITE(4,'(A,A,I3,A,3(I3,I6,I7,A))')' 1',TIPE,IAR(N,16),'  ³',(IAR
     1 (I,J),J=2,4),' ³',(IAR(I,J),J=12,14),' ³',(IAR(I,J),J=7,9)
      END IF
      END IF
C
      II=0
      DO 28 I=1,NFTNG
        IF(NTYPE(I) .EQ. 4)THEN
      II=II+1
      WRITE(4,'(/A/)')' RAFT REINFORCEMENTS'
      WRITE(4,'(/A/)')' REINFORCEMENT IN X-DIRECTION'
      WRITE(4,'(A)')'      ³          COLUMN  STRIP             ³
     1   MIDDLE  STRIP'
C
      WRITE(4,'(A)')' PANEL³     BOTTOM      ³       TOP        ³     BOT
     1TOM      ³       TOP'
      WRITE(4,'(A)')'      ³ NO   SIZE   SPA ³ NO   SIZE   SPA ³ NO   SI
     1ZE   SPA ³ NO   SIZE   SPA '
        IF(II .GT. 1)GOTO 29
        END  IF
```

```
 28   CONTINUE
 29   CONTINUE
C
      M=0
      DO 30 N=1,IRAFT
        M=M+1
          WRITE(4,'(/A,I2/)')' RAFT',M
      IPANEL=0
      NPNL=NPANEL(IRAFT)
      DO 30 I=1,NPNL
      IPANEL=IPANEL+1
      WRITE(4,'(A2,I2,A3,4(I3,I7,I7,A))')'  ',IPANEL,'  ³',(IRX(I,J),J=1
     1,3),'³',(IRX(I,J),J=5,7),'³',(IRX(I,J),J=9,11),'³',(IRX(I,J),J=13,
     215),'³'
 30   CONTINUE
C
      II=0
      DO 31 I=1,NFTNG
        IF(NTYPE(I) .EQ. 4)THEN
      II=II+1
      WRITE(4,'(/A/)')' REINFORCEMENT IN Y-DIRECTION'
      WRITE(4,'(A)')'      ³           COLUMN  STRIP            ³
     1   MIDDLE  STRIP'
C
      WRITE(4,'(A)')' PANEL³     BOTTOM      ³       TOP        ³     BOT
     1TOM      ³       TOP'
      WRITE(4,'(A)')'      ³ NO   SIZE   SPA ³ NO   SIZE   SPA ³ NO   SI
     1ZE   SPA ³ NO   SIZE   SPA '
       IF(II .GT.1)GOTO 32
        END IF
 31   CONTINUE
 32   CONTINUE
C
      M=0
      DO 40 N=1,IRAFT
        M=M+1
          WRITE(4,'(/A,I2/)')' RAFT',M
      IPANEL=0
      NPNL=NPANEL(IRAFT)
      DO 40 I=1,NPNL
      IPANEL=IPANEL+1
      WRITE(4,'(A2,I2,A3,4(I3,I7,I7,A))')'  ',IPANEL,'  ³',(IRX(I,J),J=1
     17,19),'³',(IRX(I,J),J=21,23),'³',(IRX(I,J),J=25,27),'³',(IRX(I,J),
     2J=29,31),'³'
 40   CONTINUE
C

      END
C
C
      SUBROUTINE REED(DAT)
      COMMON /A/ NCOLS,ABCAP
      COMMON /B/ XMAX,YMAX
      COMMON /D/ FOOTD(100,7),CLPR(100,12)
      COMMON /E/ IST(100),NTYPE(100),MATRS(100,3)
      COMMON /P/ PMMT(10,10),PRENF(10,8)
      COMMON /Q/ BARS(5)
      COMMON /R/ IRAF(100,9),NPANEL(10),IRX(10,32)
      COMMON IRAFT
      COMMON NFTNG
      CHARACTER DAT2
C
C Initialise arrays
C
      ICOL=0
```

```
      DO 10 I=1,100
         ICOL=ICOL+1
         IST(I)=ICOL
         ntype(I)=1
         NPANEL(I)=0
         DO 5 J=1,12
            FOOTD(I,J)=0
            CLPR(I,J)=0
            IRAF(I,J)=0
            PMMT(I,J)=0
            PRENF(I,J)=0
  5      CONTINUE
 10   CONTINUE
C
c Enter required data interactively from screen
c
      IF(S.EQ.'S')THEN
        WRITE(*,*)' HOW MANY COLUMNS DO YOU HAVE ?'
        NCOLS=1
        READ(*,*)NCOLS
        ICOL=0
        DO 30 I=1,NCOLS
           ICOL=ICOL+1
           WRITE(*,'(/A,I3,/A)')' ENTER THE FOLLOWING FOR COLUMN',ICOL,
     1     ' COORDINATES OF LOWER LEFT CORNER'
           CLPR(I,1)=25.
           CLPR(I,2)=25.
           READ(*,*)(CLPR(I,J),J=1,2)
C           WRITE(4,'(/A,I3//A,2F3.2)')' COLUMN ',ICOL,' COORDINATES OF
C     1     LOWER LEFT CORNER',(CLPR(I,J),J=1,2)
C
           WRITE(*,'(/A)')' LENGTH IN X AND Y DIRECTIONS IN mm'
           CLPR(I,3)=230.
           CLPR(I,4)=230.
           READ(*,*)(CLPR(I,J),J=3,4)
C           WRITE(4,'(A,2F6.2)')' LENGTH IN X AND Y DIRECTIONS (mm)',
C     1     (CLPR(I,J),J=3,4)
C
           WRITE(*,'(/A)')' CHARACTERISTIC DEAD AND LIVE LOADS IN kN'
           CLPR(I,5)=57.6
           CLPR(I,6)=24.
           READ(*,*)(CLPR(I,J),J=5,6)
C           WRITE(4,'(A,2F8.2)')' CHARACTERISTIC DEAD AND LIVE LOADS kN'
C     1     ,(CLPR(I,J),J=5,6)
C
           WRITE(*,'(/A)')' SIZE OF DOWEL BARS'
           MATRS(I,1)=16
           READ(*,'(I2)')MATRS(I,1)
C
           WRITE(*,'(/A)')' CUBE STRENGTH'
           MATRS(I,2)=25
           READ(*,'(I2)')MATRS(I,2)
C
           WRITE(*,'(/A)')' STEEL STRENGTH'
           MATRS(I,3)=410
           READ(*,'(I3)')MATRS(I,3)
C
  30    CONTINUE
C
  41      FORMAT(/A)
  42      FORMAT(F8.2)
  43      FORMAT(A,F8.2)
C
          WRITE(*,41)' ENTER ALLOWABLE BEARING CAPACITY'
          abcap=150.
```

```
          REad(*,*)abcap
          WRITE(*,43)' ALLOWABLE BEARING CAPACITY GIVEN AS ',abcap
C
c Read data from an already prepared data file
c
      ELSE
         READ(3,*)NCOLS
         DO 50 I=1,NCOLS
           READ(3,*)(CLPR(I,J),J=1,6)
           READ(3,*)MATRS(I,1)
 50      CONTINUE
         READ(3,*)ABCAP
         READ(3,*)fcu,fy
         do 51 i=1,ncols
          matrs(i,2)=fcu
          matrs(i,3)=fy
 51      continue
      END IF
C
      RETURN
      END
c
C
      SUBROUTINE CHOOSE
      COMMON /A/ NCOLS,ABCAP
      COMMON /E/ IST(100),NTYPE(100),MATRS(100,3)
      COMMON /P/ PMMT(10,10),PRENF(10,8)
      COMMON /Q/ BARS(5)
      COMMON /R/ IRAF(100,9),NPANEL(10),IRX(10,32)
      COMMON IRAFT
      COMMON NFTNG
      CHARACTER DAT*2
C
c Numerical identification of each type of footing
c
      WRITE(*,'(A/)')' TO CHOOSE FOOTING TYPE ENTER THE NUMBER  CORRESPO
     1NDING TO THE FOOTING :',' PAD       1',' COMBINED  2',' STRIP
     23',' RAFT      4'
C
c Instructions on how to choose footing types
c
      WRITE(*,'(2A/)')' IF ALL THE FOOTINGS ARE THE SAME TYPE ENTER 1',
     1' , ELSE ENTER THE NUMBER OF DIFFERENT TYPES'
      REad(*,'(I2)')ITYPE
      IF(ITYPE .EQ. 1)THEN
        WRITE(*,'(A)')' ENTER FOOTING TYPE'
        REad(*,'(I2)')KTYPE
        J=0
        DO 10 I=1,NCOLS
           J=J+1
           IF(IST(J) .EQ. J)THEN
             IF(KTYPE .EQ. 2)CALL CHOSE2(J)
             IF(KTYPE .EQ. 3 .OR. KTYPE .EQ. 4)CALL CHOSE3(J,KTYPE)
           END IF
10      CONTINUE
      ELSE
        J=0
        DO 20 I=1,NCOLS
           J=J+1
           IF(IST(J) .EQ. J)THEN
             WRITE(*,'(A,I3)')' ON WHAT TYPE OF FOOTING IS COLUMN ',J
             REad(*,'(I3)')KTYPE
             IF(KTYPE .EQ. 1)GOTO 20
             IF(KTYPE .EQ. 2)CALL CHOSE2(J)
             IF(KTYPE .EQ. 3      .OR. KTYPE .EQ. 4)CALL CHOSE3(J,KTYPE)
```

```
            END IF
 20       CONTINUE
          END IF
          RETURN
          END
C
          SUBROUTINE CHOSE2(J)
          COMMON /E/ IST(100),NTYPE(100),MATRS(100,3)
C
          ntype(J)=2
          WRITE(*,'(A,I2)')' WHAT COLUMN IS PAIRED WITH COLUMN',J
          REad(*,'(I2)')JCOL
c
c Change IST value for column JCOL to equal that of column J
c
          IST(JCOL)=IST(J)
          ntype(JCOL)=2
          RETURN
          END
C
C
          SUBROUTINE CHOSE3(J,KTYPE)
          COMMON /A/ NCOLS,ABCAP
          COMMON /E/ IST(100),NTYPE(100),MATRS(100,3)
          COMMON /P/ PMMT(10,10),PRENF(10,8)
          COMMON /Q/ BARS(5)
          COMMON /R/ IRAF(100,9),NPANEL(10),IRX(10,32)
          COMMON IRAFT
          COMMON NFTNG
          CHARACTER ANS,DAT*2,TIPE*5
C
c Assign value to TIPE depending on whether KTYPE is 3 (strip)
c or 4 (raft)
c
          IF(KTYPE .EQ. 3)TIPE='STRIP'
          IF(KTYPE .EQ. 4)TIPE='RAFT'
C
          WRITE(*,'(3A)')' ARE ALL THE COLUMNS ON ONE ',TIPE,' (Y OR N)'
          REad(*,'(A)')ANS
          IF(ANS .EQ. 'Y')THEN
            DO 10 I=1,NCOLS
               IF(KTYPE .EQ. 3)ntype(I)=3
               IF(KTYPE .EQ. 4)ntype(I)=4
               IST(I)=IST(J)
 10          CONTINUE
          ELSE
            IF(KTYPE .EQ. 3)ntype(J)=3
            IF(KTYPE .EQ. 4)ntype(J)=4
            WRITE(*,'(3A,I3)')' HOW MANY OTHER COLUMNS ARE ON THE SAME ',
           1  TIPE,' WITH COLUMN ',J
            REad(*,'(I3)')NC
            WRITE(*,'(A)')' ENTER THE COLUMNS ONE BY ONE'
            NN=0
            DO 15 N=1,NC
               NN=NN+1
               REad(*,*)M
               IST(M)=IST(J)
               ntype(M)=ntype(J)
               IF(NN.NE. NC)WRITE(*,'(A)')' NEXT'
 15          CONTINUE
          END IF
          RETURN
          END
C
          SUBROUTINE DESIGN(ICOL)
```

```
      COMMON /A/ NCOLS,ABCAP
      COMMON /B/ XMAX,YMAX
      COMMON /D/ FOOTD(100,7),CLPR(100,12)
      COMMON /E/ IST(100),NTYPE(100),MATRS(100,3)
      COMMON /F/ AR(99,10),IAR(99,16)
      common /h/ dl1
      COMMON /P/ PMMT(10,10),PRENF(10,8)
      COMMON /Q/ BARS(5)
      COMMON /R/ IRAF(100,9),NPANEL(10),IRX(10,32)
      COMMON IRAFT
      COMMON NFTNG
      CHARACTER DAT*2
C
C Increase number of footings by 1 every time DESIGN is called
c
      NFTNG=NFTNG+1
C
c Depending on the value of NTYPE for column ICOL call the appropriate
c subroutine to design the footing for the column, then set the value
c of NTYPE for NFTNG accordingly
c
      IF(ntype(ICOL) .EQ. 1)THEN
        CALL DESIN1(ICOL)
        NTYPE(NFTNG)=1
      ELSE IF(ntype(ICOL) .EQ. 2)THEN
        CALL DESIN2(ICOL)
        NTYPE(NFTNG)=2
      ELSE IF(ntype(ICOL) .EQ. 3)THEN
        CALL DESIN3(ICOL)
        NTYPE(NFTNG)=3
      ELSE
        IRAFT=IRAFT+1
        CALL DESIN4(ICOL)
        NTYPE(NFTNG)=4
      END IF
C
      RETURN
      END
c
C
      SUBROUTINE DESIN1(ICOL)
      COMMON /A/ NCOLS,ABCAP
      COMMON /B/ XMAX,YMAX
      COMMON /D/ FOOTD(100,7),CLPR(100,12)
      COMMON /E/ IST(100),NTYPE(100),MATRS(100,3)
      COMMON /F/ AR(99,10),IAR(99,16)
      common /h/ dl1
      COMMON /Q/ BARS(5)
      COMMON /R/ IRAF(100,9),NPANEL(10),IRX(10,32)
      COMMON IRAFT
      COMMON NFTNG
      CHARACTER DAT*2
C
c Set the value of the (NFTNG,16) element of IAR to reflect a single
c footing
c
      IAR(NFTNG,16)=1
C
c Assume a value for the thickness of the footing
c
      FT=21.*MATRS(ICOL,1)
c
c Assume a value for the diameter of bending reinforcement
c
      br=12.
```

```
C
c Intialise counters
c
      itm=0
      itf1=0
      i=1
      j=1
C
c Calculate dimensions of the footing
c
  5   xl=sqrt(((1.+dl1)*clpr(icol,5)+clpr(icol,6))/abcap)
c?      cl=((1.+dl1)*clpr(icol,5)+clpr(icol,6))
      yl=xl
c
c Calculate the coordinates of the corners of the footing
c
      X1=(CLPR(I,1)+(CLPR(I,3)-XL))*.5
      X2=X1+XL
      Y1=(CLPR(I,2)+(CLPR(I,4)-yl))*.5
      Y2=Y1+yl
C
c Calculate the actual weight of the footing
c
 6    FTWT=XL*YL*FT*1.E-3*24.
c
c Check the weight of the footing against the assumed. If the actual
c is greater than 15%, or less than 10%, of column dead load,  calculate
c the value of dl1 to give actual weight, then go back and adjust all
c affected calculations
c
      IF(FTWT*.98 .GT. dl1*CLPR(ICOL,5).or. ftwt .lt. .67*dl1*clpr(icol,
     15))THEN
        dl1=ftwt/clpr(icol,5)
        GOTO 5
      END IF
c
c Calculate effective depth
c
  10  ed=ft-50.-br
c
c Calculate earth pressure and net upward pressure
c
      EP=((1.+dl1)*CLPR(ICOL,5)*1.4+1.6*CLPR(ICOL,6))/XL/YL
      UPN=EP-FT*33.6*1.E-3
C
c Set the values of CLPR(ICOL,3) and CLPR(ICOL,4) to XLC and YLC
c for easier handling
c
      XLC=CLPR(ICOL,3)
      YLC=CLPR(ICOL,4)
c
c Calculate the values of critical perimeter, area within perimeter
c and punching shear force
c
      CP=(2.*(XLC+YLC)+9.42*FT)*1.E-3
      AWP=(XLC+3.*FT)*(YLC+3.*FT)*1.E-6-.86*(1.5E-3*FT)**2
      PSF=UPN*(XL*YL-AWP)
c
c If PSF is negative set it to 0
c
      IF(PSF .LT. 0.)PSF=0.
c
c Calculate punching shear stress
c
      PSS=PSF/CP/ED
```

```
c
c Calclate wide beam shears in X and Y directions
c
      SSX=(((.5*(YL*1.E3-YLC))-1.5*ED))*UPN/ED*1.E-3
      SSY=(((.5*(XL*1.E3-XLC))-1.5*ED))*UPN/ED*1.E-3
c
c If wide beam shears are negative set them to 0
c
      IF(SSX .LT. 0.)SSX=0.
      IF(SSY .LT. 0.)SSY=0.
C
c Calculate bending moments about X and Y axes
c
      BMY=UPN*YL*.125*(XL-XLC*1.E-3)**2
      BMX=UPN*XL*.125*(YL-YLC*1.E-3)**2
c
c Calculate beta
c
      BT=.15*MATRS(ICOL,2)*ED*ED
c
c Check that moments are not greater than section ultimate moments
c If they are increase footing thickness by 10% and go back to adjust
c all affected calculations
c
      IF(BMY*1.E6 .GT.BT*XL*1.E6 .OR. BMX*1.E6 .GT. BT*YL*1.E6)THEN
        FT=FT*1.1
        itm=itm+1
        if(itm .gt. 98)stop
        GOTO 6
      END IF
c
c Determine steel ratio in both directions, ROX and ROY
C
      BY=BMY/XL/ED/ED*1.E3
      BX=BMX/YL/ED/ED*1.E3
      ROX=.46*MATRS(ICOL,2)/MATRS(ICOL,3)*(1.-(1.-5.*Bx/MATRS(ICOL,2))**
     1.5)
      ROY=.46*MATRS(ICOL,2)/MATRS(ICOL,3)*(1.-(1.-5.*By/MATRS(ICOL,2))**
     1.5)
      IF(MATRS(ICOL,3) .GT. 250)THEN
        ROX=MAX(ROX,.0015)
        ROY=MAX(ROY,.0015)
      ELSE
        ROX=MAX(ROX,.0025)
        ROY=MAX(ROY,.0025)
        ROA=ROX
      END IF
      IF(ROX .GT. .03)ROX=.03
      IF(ROY .GT. .03)ROY=.03
C
c Calculate ultimate shear values
c
      FD=400./ED
      IF(FD .LT. 1.)FD=1.
      IF(MATRS(ICOL,2) .LE. 25 .OR. MATRS(ICOL,2).GT. 40 )RK2=1.
      IF(MATRS(ICOL,2) .GT. 25 .AND. MATRS(ICOL,2) .LE. 40)RK2=(MATRS(
     1ICOL,2)/25)**.33
      USSX=RK2*.79*(ROX*100.)**.33*FD**.25/1.25
      USSY=RK2*.79*(ROY*100.)**.33*FD**.25/1.25
c
c Check that shears do not exceed ultimate values. If they do increase
c footing thickness by 10% and go back to adjust all affected
calculations
c
      IF(PSS .GT.                MIN(USSX,USSY))THEN
```

```
        FT=FT*1.1
        GOTO 6
      END IF
      IF(SSX.GT.USSX .OR. SSY.GT.USSY)THEN
        FT=FT*1.1
        GOTO 6
      END IF
C
C Calculate areas of bending reinforcement
C
      ASX=ROX*ED*XL*1000.
      ASY=ROY*ED*YL*1000.
C
C Assume diameter  of  bending reinforcement as brx and bry. With i
C and j equal to 1, initial assumption for both reinforcements is 12.
C
      BRX=BARS(I)
      BRY=BARS(J)
C
C Calculate cross-section area of bars
C
      ABRX=22.*BRX**2/28.
      ABRY=22.*BRY**2/28.
C
C Calculate number of bars and round to the nearest integer.
C
      BRNX=INT((ASX/ABRx)+.5)
      BRNY=INT((ASY/ABRy)+.5)
C
C Calculate wide beam shear
C
      V1X=UPN*YL*.5*(XL-XLC*1.E-3)
      V1Y=UPN*XL*.5*(YL-YLC*1.E-3)
C
C Calculate local bond stress
C
      BSLX=V1X/(BRNX*ABRX*ED)*1.E3
      BSLY=V1Y/(BRNY*ABRY*ED)*1.E3
C
C Calculate ultimate bond  stress
C
      UBS=.28*MATRS(ICOL,2)**.5
C
C Adjust ultimate bond stress for FY greater than 250
C
      IF(MATRS(ICOL,3) .GT. 250.)UBS=.6*MATRS(ICOL,2)**.5
C
C Check that local bond stresses do not exceed ultimate. If they do
C increase footing thickness by 10% and go back  to adjust all affected
C calculations.
C
      IF(BSLX .GT. UBS .or. BSLY .GT. UBS)THEN
        FT=FT*1.1
        GOTO 6
      END IF
C
C Calculate spacing of reinforcement
C
      SX=INT(XL/(BRNx-1.)*1.E3/25.)*25.
      SY=INT(YL/(BRNy-1.)*1.E3/25.)*25.
C
C Check that spacing of bars does not exceed 300. If it does set
C it to 300
C
      IF(SX .GT. 300.)SX=300.
```

```
c
c Check that spacing of bars is not less than 175. If it is, use the
c next bar size
c
      IF(SX .LT. 175.)THEN
        I=I+1
        BRX=BARS(I)
c
c If after using the fifth bar size spacing is still less than  175

c increase footing thickness by 10% and go back to adjust all affected
c calculations
c
        if(i .gt. 5)then
          i=1
          ft=ft*1.1
          brx=bars(i)
          goto 6
        end if
        GOTO 10
      end if
c
c Adjust calculations affected by change of bar size
c
      BRNX=INT(XL*1.E3/SX+.5)+1.
      ASXP=ABRX*BRNX
      XL=((BRNX-1.)*SX)*1.E-3
      ROX=ASXP/XL/ED*1.E-1
c
c Do the same checks on spacing cf reinforcement in  Y-direction
c
      IF(Sy .GT. 300.)Sy=300.
      if(sy .lt. 175.)then
        j=j+1
        bry=bars(j)
        if(j .gt. 5)then
          j=1
          ft=ft*1.1
          bry=bars(j)
          goto 6
        end if
        GOTO 10
      end if
      BRNy=INT(yL*1.E3/sy+.5)+1.
      ASYP=BRNY*ABRY
      YL=((BRNY-1.)*SY)*1.E-3
      ROY=ASYP/YL/ED*1.E-1
C
c Calculate weight of reinforcement, volume of concrete and weight of
c reinforcement per cubic meter of concrete for the footing
c
      WT=.00786*(ABRX*BRNX*XL+ABRY*BRNY*YL)
      VOL=FT*XL*YL*1.E-3
      WTV=WT/FT/XL/YL*1.E3
C
c Store calculated values in the appropriate arrays
c
      K=NFTNG
      AR(K,1)=FT
      AR(K,2)=XL
      AR(K,3)=YL
      AR(K,4)=X1
      AR(K,5)=X2
      AR(K,6)=Y1
      AR(K,7)=Y2
```

```
      AR(K,8)=VOL
      AR(K,9)=WT
      AR(K,10)=WTV
C

      IAR(K,1)=INT(ASXP)
      IAR(K,2)=INT(BRNX)
      IAR(K,3)=INT(BRX)
      IAR(K,4)=INT(SX)
      IAR(K,5)=INT(ROX/100.)
      IAR(K,6)=INT(ASYP)
      IAR(K,7)=INT(BRNY)
      IAR(K,8)=INT(BRY)
      IAR(K,9)=INT(SY)
      IAR(K,10)=INT(ROY/100.)
c
c Set the elements  representing top reinforcement to 0
c
      DO 15 I=11,15
        IAR(K,I)=0
 15   CONTINUE
C
      RETURN
      END
C
      SUBROUTINE DESIN2(ICOL)
      COMMON /A/ NCOLS,ABCAP
      COMMON /B/ XMAX,YMAX
      COMMON /D/ FOOTD(100,7),CLPR(100,12)
      COMMON /E/ IST(100),NTYPE(100),MATRS(100,3)
      COMMON /F/ AR(99,10),IAR(99,16)
      common /h/ dl1
      COMMON /Q/ BARS(5)
      COMMON /R/ IRAF(100,9),NPANEL(10),IRX(10,32)
      COMMON IRAFT
      COMMON NFTNG
      CHARACTER ANS,DAT*2
C
c Set the value of the (NFTNG,16) element of IAR to reflect a combined
c footing
c
      IAR(NFTNG,16)=2
C
c Identify the column that is combined with column JCOL and designate
c as column K
c
      J=ICOL
      DO 10 I=ICOL+1,NCOLS
        IF(IST(I) .EQ. IST(ICOL))K=I
        IF (K.EQ.I)GOTO 11
 10   CONTINUE
 11   CONTINUE
C
c Set the sum of the dead loads of the two columns to DL and the sum
c of live loads to VL
c
      DL=CLPR(J,5)+CLPR(K,5)
      VL=CLPR(J,6)+CLPR(K,6)
c
c dl2  makes provision for footing weight when determining
c soil pressure. It is initially taken as 10%
c
      DL2=0.1
c
c  Assume a value for the         thickness of the footing
```

```
c
      FT =21.*MATRS(ICOL,1)
C
c Calculate the coordinates (xcj,ycj) and (xck,yck) of centers of
c columns J and K
c
      XCJ=CLPR(J,1)+(CLPR(J,3)*1.e-3)*.5
      YCJ=CLPR(J,2)+(CLPR(J,4)*1.e-3)*.5
      XCK=CLPR(K,1)+(CLPR(K,3)*1.e-3)*.5
      YCK=CLPR(K,2)+(CLPR(K,4)*1.e-3)*.5
C
      XLF=ABS(XCJ-XCK)
      YLF=ABS(YCJ-YCK)
      D=XLF+YLF
c
c Calculate the area and dimensions the footing
c
 12   AREA=(DL*(1.+DL2)+VL)/ABCAP
      XL=XLF/D*AREA
      YL=YLF/D*AREA
C
c Check that either dimension is at least half the other
c
      IF(XL .LT. AREA/SQRT(AREA/.5) )THEN
        XL=AREA/SQRT(AREA/.5)
        YL=AREA/XL
      ELSE IF(YL .LT. AREA/SQRT(AREA/.5) )THEN
        YL=AREA/SQRT(AREA/.5)
        XL=AREA/YL
      END IF
C
c Determine parameters for column to the left and column to right
c
      IF(XCJ .LT. XCK)THEN
        XCL=XCJ
        YCL=YCJ
        XCR=XCK
        YCR=YCK
        XLCL=CLPR(J,3)*1.e-3
        YLCL=CLPR(J,4)*1.e-3
        XLCR=CLPR(K,3)*1.e-3
        YLCR=CLPR(K,4)*1.e-3
      ELSE IF(XCJ .GE. XCK)THEN
        XCL=XCK
        YCL=YCK
        XCR=XCJ
        YCR=YCJ
        XLCL=CLPR(K,3)*1.e-3
        YLCL=CLPR(K,4)*1.e-3
        XLCR=CLPR(J,3)*1.e-3
        YLCR=CLPR(J,4)*1.e-3
      END IF
C
c Determine parameters for column to above and column below
c
      IF(YCJ .LT. YCK)THEN
        YCB=YCJ
        XCB=XCJ
        YCA=YCK
        XCA=XCK
        YLCB=CLPR(J,4)*1.e-3
        XLCB=CLPR(J,3)*1.e-3
        YLCA=CLPR(K,4)*1.e-3
        XLCA=CLPR(K,3)*1.e-3
      ELSE IF(YCJ .GE. YCK)THEN
```

```
        YCB=YCK
        XCB=XCK
        YCA=YCJ
        XCA=XCJ
        YLCB=CLPR(K,4)*1.e-3
        XLCB=CLPR(K,3)*1.e-3
        YLCA=CLPR(J,4)*1.e-3
        XLCA=CLPR(J,3)*1.e-3
      END  IF
C
c Set the bigger dimension of XL and YL as BD, the smaller as SD
c
      IF(XCR-XCL .GT. YCA-YCB)THEN
        BD=XL
        SD=YL
      ELSE
        BD=YL
        SD=XL
      END IF
C
c If XL is BD calculate  BAR as distance of base centroid from
c left edge
c
      IF(BD .EQ. XL)THEN
        IF(XLCL .EQ. CLPR(J,3)*1.e-3)THEN
          I=J
        ELSE
          I=K
        END IF
        BAR=((CLPR(I,5)+CLPR(I,6))*(XCR-XCL))/(DL+VL)
c
c Calculate coordinates of base centroid
c
        XR=XCR-BAR
        YR=YCB+.5*(YCA-YCB)
c
c If YL is BD calculate  BAR as distance of base centroid from
c bottom edge
c
      ELSE
        IF(YLCB .EQ. CLPR(J,4)*1.e-3)THEN
          I=J
        ELSE
          I=K
        END IF
        BAR=((CLPR(I,5)+CLPR(I,6))*(YCA-YCB))/(DL+VL)
c
c Calculate coordinates of base centroid
c
c
        YR=YCA-BAR
        XR=XCL+.5*(XCR-XCL)
      END IF
C
c Calculate coordinates of base corners
c
      X1=XR-.5*XL
      X2=X1+XL
      Y1=YR-.5*YL
      Y2=Y1+YL
c
c Check that distance from edge of base to nearest column is not
c less than 1.5 times the dimension in that direction
c
      IF(XCL-X1 .LT.              1.5*XLCL)X1=XCL-1.5*XLCL
```

```
      IF(X2-XCR .LT. 1.5*XLCR)X2=XCR+1.5*XLCR
      HXL=MAX((XR-X1),(X2-XR))
      IF(XR-X1 .LT. HXL)X1=XR-HXL
      IF(X2-XR .LT. HXL)X2=XR+HXL
      XL=X2-X1
C
      IF(YCB-Y1 .LT. 1.5*YLCB)Y1=YCB-1.5*YLCB
      IF(Y2-YCA .LT. 1.5*YLCA)Y2=YCA+1.5*YLCA
      HYL=MAX((YR-Y1),(Y2-YR))
      IF(YR-Y1 .LT. HYL)Y1=YR-HYL
      IF(Y2-YR .LT. HYL)Y2=YR+HYL
      YL=Y2-Y1
C
c Calculate area using new dimensions.
c
      AREA=XL*YL
c
      IF(XCR-XCL .GT. YCA-YCB)THEN
        BD=XL
        SD=YL
      ELSE
        BD=YL
        SD=XL
      END IF
C
c Calculate effective depth
c
  15  ED=FT-60.
c
c Calculate the actual weight of the footing
c
      ftwt=xl*yl*ft*24.e-3
c
c Check the weight of the footing against the assumed. If the actual
c is greater than 10%, or less than 8%, of column dead load,  calculate
c the value of dl1 to give actual weight, then go back and adjust all
c affected calculations
c
      if(ftwt*.98 .gt. dl2*dl .or. ftwt .lt. .8*dl2*dl)then
        dl2=ftwt/dl
        GOTO 12
      end if
C
c Calculate ultimate load
c
      UL=1.4*((1.+DL2)*(CLPR(J,5)+CLPR(K,5)))+1.6*(CLPR(J,6)+CLPR(K,6))
c
c Calculate earth pressure and net upward pressure
c
      EP=UL/AREA
      UPN=EP-1.4*FT*24.e-3
C
c Determine which of XL and yl is BD then set the Z variables
c used to determine which column parameters to use in calculting
c shear force and bending moments
c
      IF(BD .EQ. XL)THEN
        Z1=X1
        Z2=X2
        ZCL=XCL
        ZLCL=XLCL
        ZCR=XCR
        ZLCR=XLCR
      ELSE
        Z1=Y1
```

```
        Z2=Y2
        ZCL=YCB
        ZLCL=YLCB
        ZCR=YCA
        ZLCR=YLCA
      END IF
C
c  Calculate equivalent net upward pressure per meter length of base
c
      W=UPN*SD
C
c Calculate reactions due to UPN under left and right columns
c
      RL=(W*BD*bar)/(ZCR-ZCL)
      RR=W*BD-RL
c
c Set the bigger reaction  as RXN
c
      RXN=MAX(RR,RL)
c
c Set column with larger load as LL, the other as LS
c
      IF(CLPR(J,5)+CLPR(J,6).GT. CLPR(K,5)+CLPR(K,6))THEN
        LL=J
        LS=K
        ZXRCC=XCJ
        ZYRCC=YCJ
      ELSE
        LL=K
        LS=J
        ZXRCC=XCK
        ZYRCC=YCK
      END IF
C
      IF(BD .EQ. XL)THEN
        IF(RXN .EQ. RR)D=Z2-ZXRCC
        IF(RXN .EQ. RL)D=ZXRCC-Z1
      ELSE
        IF(RXN .EQ. RR)D=Z2-ZYRCC
        IF(RXN .EQ. RL)D=ZYRCC-Z1
      END IF
C
c Calculate shear force under column with larger load
c
      SF=RXN-W*D
c
c Calculate shear force at distance 1.5ED from face of column
c with larger load
c
      Vd=SF-W*(1.5e-3*ED+CLPR(LL,3)*.5e-3)
c
c Calculate shear stress
c
      SSd=Vd/SD/ED
c
c Calculate coefficients for use in solving point of contraflexure
c
      A=W
      B=2.*RL
      C=B*(ZCL-Z1)
c
c Calculate point of maximum span moment
c
      XX=RL/W
c
```

```
c Calculate span moment
c
      TM=-W*XX**2*.5+RL*(XX+z1-zCL)
c
c If span moment is sagging do not provide top reinforcement
c
      IF(TM .LE. 0.)THEN
        TM=0.
        VF=0.
c
c If span moment is hogging calculate point of contraflexure and
c shear force there
c
      ELSE
        CF=((B+SQRT(B**2-4.*A*C))/(2.*A))+z1
        VF=RR-W*(Z2-CF)
      END IF
c
c Calculate distance from edge of base to face of column with
c larger load
c
      dc=d-clpr(11,3)*.5e-3
c
c Calculate shear force  at the face of column with larger load
c
      vc=upn*sd*dc
c
c Calculate moment under columns to the left and to the right
c
      RLM=.5*W*(ZCL-Z1-.5*ZLCL)**2
      RM=.5*W*(Z2-ZCR-.5*ZLCR)**2
c
c Set bottom design moment as the bigger of the two above
c
      BMMT=MAX(RM,RLM)
c
c Calculate transvers moment
c
      TRM=UPN*(.5*SD)**2*.5
c
c Check that moments are not greater than section ultimate moments
c If they are increase footing thickness by 20% and go back to adjust
c all affected calculations
c
      if(max(TM,BMMT,TRM) .gt. .15*MATRS(ICOL,2)*ed**2*bd)then
        FT=FT*1.2
        goto 15
      end if
c
c Calculate area of bottom reinforcement
c
      ASB=BMMT/(.83*MATRS(ICOL,3)*ED)*1.E6
c
c Check that ASB is not less than minimum allowed
c
      asmin=1.5*SD*ED
      IF(ASB .LT. 1.5*SD*ED)ASB=1.5*SD*ED
c
      i=1
c
c Assume diameter  of  bottom reinforcement as brb . With i
c equal to 1, intial assumption for the reinforcement is 12.
c
 23   brb=bars(i)
c
```

```
c Calculate number of bars and round to the nearest integer.
c
      NBB=INT(ASB/(.79*BRB**2)+.5)
c
c Calculate spacing of bottom bars
c
      sbb=sd*1.e3/float(nbb)-1.
c
c Check that spacing of bars is not less than 175. If it is, use the
c next bar size
c
      if(sbb .lt. 175.)then
        i=i+1
        goto 23
c
c Check that spacing of bars is not greater than 300. If it is set
c it to 300 and adjust all affected calculations
c
      else if(sbb .gt. 300.)then
        sbb=300.
        nbb=int(sd/300.+1.)
        ASB=float(nbb)*BRB**2*.79
      end if
c
c Calculate steel ratio
c
      rob = ASB/sd/ed*1.e-1
c
c Calculate local bond stress for bottom bars
c
      BSLB=Vc/(FLOAT(NBB)*3.14*BRB*ED)*1.e3
c
c Calculate ultimate bond stress
c
      UBS=.28*MATRS(ICOL,2)**.5
c
c Adjust ultimate bond stress for FY greater than 250
c
      IF(MATRS(ICOL,3) .GT. 250.)UBS=.6*MATRS(ICOL,2)**.5
c
c Check that BSLB is not greater than UBS. If it is increase footing
c thickness by 20% and adjust all affected calculations
c
      IF(BSLB.GT. .6*MATRS(ICOL,2)**.5)then
        ft=ft*1.2
        goto 15
      end if
c
c Calculate area of top reinforcement
c
      ASt=TM*1.E6/(.83*MATRS(ICOL,3)*ED)
c
c Check that AST is not less than minimum allowed
c
      IF(AST .LT. ASMIN .AND. AST .NE. 0.)AST=ASMIN
c
      i=1
c
c Assume diameter  of  top reinforcement as brt . With i
c equal to 1, initial assumption for the reinforcement is 12.
C
      i=1
 25   brt=bars(i)
c
c Calculate number of bars and    round to the nearest integer.
```

```
c
      NTB=INT(ASt/(.79*BRt**2)+.5)
c
c If top reinforcement is required check minimum and maximum
c spacing requirements
c
      IF(TM .GT. 0.)THEN
        stb=sd*1.e3/float(ntb)-1.
        if(stb .lt. 175.)then
          i=i+1
          goto 25
        else if(stb .gt. 300.)then
          stb=300.
          ntb=int(sd/300.+1.)
          ast=float(ntb)*brt**2*.79
        end if
c
c If top reinforcement is not required assign nominal values to
c number, spacing and diameter of top bars
c
      ELSE
        NTB=1
        STB=.001
        BRT=.001
      END IF
c
c Calculate steel ratio
c
      rot=ast/sd/ed*1.e-1
c
c Calculate local bond stress for top bars
c
      BSLT=VF/(FLOAT(NTB)*3.14*BRt*ED)*1.e3
c
c Check that BSLT is not greater than UBS. If it is increase footing
c thickness by 20% and adjust all affected calculations
c
      IF(BSLT .GT. .6*MATRS(ICOL,2)**.5)then
        ft=ft*1.2
        goto 15
      end if
c
c Assume diameter  of  transverse reinforcement as brtr. With i
c equal to 1, intial assumption for the reinforcement is 12.
c
      i=1
 26   brtr=bars(i)
c
c Calculate area of transverse reinforcement
c
      ASTR=TRM*1.E6/(ED*MATRS(ICOL,3)*.83)
c
c Check that ASTR is not less than minimum allowed
c
      IF(ASTR .LT. 1.5*BD*ED)ASTR=1.5*BD*ED
c
c Calculate number of bars and round to the nearest integer.
c

      NBTR=INT(ASTR/(.79*BRtr**2)+.5)
c
c Check minimum and maximum spacing requirements
c
      str=bd*1.e3/float(nbtr)-1.
      if(str .lt. 175. )then
```

```
        i=i+1
        goto 26
      else if(str .gt. 300.) then
       str=300.
       nbtr=int(bd*1.e3/300.+1.)
       astr=float(nbtr)* .79*brtr**2
      end if
c
c Calculate steel ratio
c
      rotr=astr/bd/ed*1.e-1
c
c Calculate ultimate shear
c
      FD=400./ED
      IF(FD .LT. 1.)FD=1.
      IF(MATRS(ICOL,2) .LE. 25 .or. MATRS(ICOL,2) .gt. 40)RK2=1.
      IF(MATRS(ICOL,2) .GT. 25 .AND. MATRS(ICOL,2) .LE. 40)RK2=(MATRS(
     1ICOL,2)/25)**.33
      RO=MAX(ROT,ROB,ROTR)
      USS=RK2*.79*RO**.33*FD**.25/1.25
c
c Check that shears do not exceed ultimate value. If they do increase
c footing thickness by 10% and go back to adjust all affected
calculations
c
      IF(SSD .GE. USS)THEN
        IF(SSD .GT. USS)FT=FT*1.2
        IF(SSD .EQ. USS)FT=FT*1.05
        GOTO 15
      END IF
c
      k=NFTNG
c
c Calculate cross sectional areas of reinforcements
c
      ABRB=22.*BRB**2/28.
      ABRT=22.*BRT**2/28.
      ABRTR=22.*BRTR**2/28.
C
c Store output in arrays AR and IAR
c
      AR(K,1)=FT
      AR(K,2)=XL
      AR(K,3)=YL
      AR(K,4)=X1
      AR(K,5)=X2
      AR(K,6)=Y1
      AR(K,7)=Y2
      AR(K,8)=XL*YL*FT*1.E-3
      AR(K,9)=.00786*(ABRB*FLOAT(NBB)*BD+ABRT*FLOAT(NTB)*BD+ABRTR*FLOAT(
     1NBTR)*SD)
      AR(K,10)=AR(K,9)/AR(K,8)
C
      IAR(NFTNG,1)=ASB
      IAR(NFTNG,2)=NBB
      IAR(NFTNG,3)=INT(BRB)
      IAR(NFTNG,4)=INT(SBB)
      IAR(NFTNG,5)=INT(ROB*100.)
      IAR(NFTNG,6)=INT(ASTR)
      IAR(NFTNG,7)=NBTR
      IAR(NFTNG,8)=INT(BRTR)
      IAR(NFTNG,9)=INT(STR)
      IAR(NFTNG,10)=INT(ROTR*100.)
      IAR(NFTNG,11)=INT(AST)
```

```
      IAR(NFTNG,12)=NTB
      IAR(NFTNG,13)=INT(BRT)
      IAR(NFTNG,14)=INT(STB)
      IAR(NFTNG,15)=INT(ROT*100.)
      RETURN
      END
C
C
C
      SUBROUTINE DESIN3(ICOL)
      COMMON /A/ NCOLS,ABCAP
      COMMON /D/ FOOTD(100,7),CLPR(100,12)
      COMMON /E/ IST(100),NTYPE(100),MATRS(100,3)
      COMMON /F/ AR(99,10),IAR(99,16)
      COMMON /Q/ BARS(5)
      COMMON /R/ IRAF(100,9),NPANEL(10),IRX(10,32)
      COMMON IRAFT
      COMMON NFTNG
      CHARACTER ANS,DAT*2
C
c Initialise number of columns on footing NFTNG
c
      IAR(NFTNG,16)=0
c
c dl4  makes provision for footing weight when determining
c soil pressure. It is initially taken as 20%
c
      DL4 = 0.2
c
c Assume a diameter for bending reinforcement to be used when
c calculating effective depth
c
      ROD = 16.
C
c Initialise counters: ILAST for last column on the strip, TLOAD
c TLOAD for total service load, ULOAD for ultimate load.
c
      ILAST=0
      TLOAD = 0.
      ULOAD = 0.
C
c Identify column with heaviest load on strip as IMAX,  last
c column as IL, and calculate number of columns on the strip.
c
      IC=icol-1
      DO 10 I=icol,NCOLS
          IC=IC+1
          IF(IST(I) .EQ. IST(ICOL))THEN
           IF(CLPR(I,5)+CLPR(I,6) .GE. TLOAD)THEN
            IMAX=IC
            TLOAD=CLPR(I,5)+CLPR(I,6)
            ULOAD=1.4*CLPR(I,5)+1.6*CLPR(I,6)
           END IF
           IL=IC
           IAR(NFTNG,16)=IAR(NFTNG,16)+1
          END IF
 10   CONTINUE
C
c Initialise coordinates of strip corners
c
      XA=1000.
      YA=1000.
      XD=0.
      YD=0.
c
```

```
c Determine co-ordinates of strip corners as co-ordinates of edge
c columns
c
      DO 15 I=icol,NCOLS
        IF(IST(I) .EQ. IST(ICOL))THEN
          XA=MIN(XA,CLPR(I,1))
          YA=MIN(YA,CLPR(I,2))
          XD=(MAX(XD,CLPR(I,1)+CLPR(I,3)*1.e-3))
          YD=(MAX(YD,CLPR(I,2)+CLPR(I,4)*1.e-3))
        END IF
 15   CONTINUE
c
c Move co-ordinates of strip corners 1 m away from edge columns
c
      XA=XA-1.
      YA=YA-1.
      XD=XD+1.
      YD=YD+1.
      XL=XD-XA
      YL=YD-YA
C
C Set XL as length along columns
C
      XL=MAX(XL,YL)
      YL=MIN(XL,YL)
c
c Calculate initial assumption of footing thickness
c
      FT=21.*MATRS(IMAX,1)
c
c Calculate column centres as distance between last column and
c the one preceding it
c
      CTRS=MAX((CLPR(Il,1)-CLPR(Il-1,1)),(CLPR(Il,2)-CLPR(Il-1,2)))
C
c Calculate strip width, YL
c
 20   YL=(1.+DL4)*TLOAD/(ABCAP*CTRS)
C
c Calculate effective depth
c
 25   ED=FT-40.-(.5*ROD)
C
c Calculate actual footing weight
c
      FTWT=CTRS*YL*FT*24.E-3
c
c Check the weight of the footing against the assumed. If the actual
c is greater than, or less than 80%, of assumed weight,  calculate
c the value of dl4 to give actual weight, then go back and adjust all
c affected calculations
c
      IF(FTWT.GT.DL4*TLOAD.OR.FTWT.LT..8*DL4*TLOAD)THEN
        IF(DL4 .EQ. FTWT/TLOAD)GOTO 26
        DL4=FTWT/TLOAD
        GOTO 20
      END IF
 26   CONTINUE
C
c Calculate earth pressure and net upward pressure
c
      EP=(1.4*(CLPR(IMAX,5)+FTWT) + 1.6*CLPR(IMAX,6))/(YL*CTRS)
      UPN=EP-1.4E-3*FT*24.
C
C  Calculate moment under         columns (FL/10)
```

```
C
      PM= UPN*YL*CTRS**2/10.
c
c Calculate area of bottom reinforcement and check that it is not
c less than minimum allowed
c
      ASB=(PM*1.E6)/(.83*MATRS(IMAX,3)*ED)
      IF(ASB .LT. 1.5*YL*ED)ASB=1.5*YL*ED
c
      IB=1
c
c Assume diameter  of bottom reinforcement as BRB. With IB
c equal to 1, initial assumption for the reinforcement is 12.
c
 50   BRB=BARS(IB)
c
c Calculate number of bottom bars and round to the nearest integer.
c
      NBB=INT((ASB/(.79*BRB**2)+.5))
c
c Calculate spacing of bottom bars
c
      ISB=INT(YL*1.E3/(FLOAT(NBB)-1.)/25.)*25
c
c Check that spacing is not less than 150. If it is use the next
c bar size
c
      IF(ISB .LT. 150)THEN
        IB=IB+1
        GOTO 50
      END IF
C
C  Check  bar diameter used in calculating ED
C
        IF(ROD .NE. BRB)THEN
          ROD=BRB
          GOTO 25
        END IF
C
c Assume diameter for calculating effective depth for span reinforcement
C
      RODS=16.
c
c Calculate effective depth for span reinforcement
c
 54   EDS=FT-40.-(.5*RODS)
c
c Calculate span moment
c
      TM=UPN*YL*CTRS**2/14.
c
c Calculate area of span reinforcement
c
      AST=TM*1.E6/(.83*MATRS(IMAX,3)*EDS)
c
c Check that AST is not less than minimum allowed
c
      IF(AST .LT. 1.5*YL*EDS)AST=1.5*YL*EDS
      IB=1
c
c Assume diameter  of span reinforcement as BRT. With IB
c equal to 1, initial assumption for the reinforcement is 12.
c
 60   BRT=BARS(IB)
C
```

```
c Calculate number of span bars and round to the nearest integer.
c
      NTB=INT((AST/(.79*BRT**2)+.5))
c
c Calculate spacing of top bars
c
      ISTB=INT(YL*1.E3/(FLOAT(NTB)-1.)/25.)*25
c
c Check that spacing is not less than 150. If it is use the next
c bar size
c
      IF(ISTB .LT. 150)THEN
        IB=IB+1
        GOTO 60
      END IF
c
C  Check  bar diameters used in calculating ED
C
        IF(RODS .NE. BRT)THEN
          RODS=BRT
          GOTO 54
        END IF
C
C  Calculate transverse moment( TRM IS PER M) and everything
c  else as for top reinforcement
C
      RODTR=16.
 61   EDTR=FT-40.-BRT-BRB-(.5*RODTR)
      TRM=UPN*(YL*.5)**2*.5
      ASTR=TRM*1.E6/(.83*MATRS(IMAX,3)*EDTR)
      IF(ASTR .LT. 1.5*EDTR)ASTR=1.5*EDTR
      IB=1
 65   BRTR=BARS(IB)
      NTRB=INT((ASTR/(.79*BRTR**2)+.5))
      ISTR=INT(1.E3/(FLOAT(NTRB)-1.)/25.)*25
      IF(ISTR .LT. 150)THEN
        IB=IB+1
        GOTO 65
      END IF
      IF(RODTR .NE. BRTR)THEN
          RODTR=BRTR
          GOTO 61
      END IF
C
C  Calculate shear and carry out checks as for a combined footing
C
        SF=UPN*YL*(CTRS*.5-(1.5E-3*ED)-.5E-3*CLPR(IMAX,3))
        SS=SF/YL/ED
        FD=400./ED
        IF(FD .LT. 1.)FD=1.
        IF(MATRS(1,2) .LE. 25)RK2=1.
        IF(MATRS(1,2) .GT. 25 .AND. MATRS(1,2) .LE. 40)RK2=(MATRS(1,2)/2
     1  5)**.33
        RO=100.*ASB/(YL*1.E3)/ED
        USS=RK2*.79*(RO)**.33*FD**.25/1.25
        IF(SS .GT. USS)THEN
          FT=FT*1.02
          GOTO 25
        END IF
 55   CONTINUE
C
c Store output in arrays AR and IAR
c
      ABRB=22.*BRB**2/28.
      ABRT=22.*BRT**2/28.
```

```
      ABRTR=22.*BRTR**2/28.
      K=NFTNG
      AR(K,1)=FT
      AR(K,2)=XL
      AR(K,3)=YL
      AR(K,4)=XA
      AR(K,5)=XD
      AR(K,6)=YA
      AR(K,7)=YD
      AR(K,8)=XL*YL*FT*1.E-3
      AR(K,9)=.00786*(ABRB*FLOAT(NBB)*XL+ABRT*FLOAT(NTB)*XL+ABRTR*FLOAT(
     1NTRB)*YL)
      AR(K,10)=AR(K,9)/AR(K,8)
C
      IAR(NFTNG,1)=ASB
      IAR(NFTNG,2)=NBB
      IAR(NFTNG,3)=INT(BRB)
      IAR(NFTNG,4)=ISB
      IAR(NFTNG,5)=ROB
      IAR(NFTNG,6)=INT(ASTR)
      IAR(NFTNG,7)=NTRB
      IAR(NFTNG,8)=BRTR
      IAR(NFTNG,9)=ISTR
      IAR(NFTNG,10)=ROTR
      IAR(NFTNG,11)=AST
      IAR(NFTNG,12)=NTB
      IAR(NFTNG,13)=BRT
      IAR(NFTNG,14)=ISTB
      IAR(NFTNG,15)=ROT
      RETURN
      END
C
C
      SUBROUTINE DESIN4(ICOL)
C
      COMMON /A/ NCOLS,ABCAP
      COMMON /D/ FOOTD(100,7),CLPR(100,12)
      COMMON /E/ IST(100),NTYPE(100),MATRS(100,3)
      COMMON /F/ AR(99,10),IAR(99,16)
      COMMON /P/ PMMT(10,10),PRENF(10,8)
      COMMON /Q/ BARS(5)
      COMMON /R/ IRAF(100,9),NPANEL(10),IRX(10,32)
      COMMON IRAFT
      COMMON NFTNG
      DIMENSION IPC(100,4)
      CHARACTER ANS,DAT*3,NAME,TOREAD
C
c Initialise the three variables below. TOREAD determines if input
c data is to be read or not in case the program goes through a GOTO
c statement.
c
      TOREAD='Y'
      DL4 = 0.2
      ROD = 16.
C
      TLOAD = 0.
      ULOAD = 0.
      XA=1000.
      YA=1000.
      XD=0.
      YD=0.
      PRENF(I,1)=0.

C
c Initialise the array IPC,       which stores the columns at the
```

```
corners of
c each panel.
c
      DO 5 I=1,100
      DO 5 J=1,4
      IPC(I,J)=0
  5   CONTINUE
C
      DO 10 I=1,NCOLS
        IF(IST(I).EQ.IST(ICOL))THEN
c
c         Calculate the total service and ultimate loads from the
columns
c         on the same raft with column "I"
c
          TLOAD=TLOAD+CLPR(I,5)+CLPR(I,6)
          ULOAD=ULOAD+1.4*CLPR(I,5)+1.6*CLPR(I,6)
c
c         Increase the number of columns on raft "NFTNG" by one.
c
          IAR(NFTNG,1)=IAR(NFTNG,1)+1
c
c         Determine the co-ordinates of the 4 columns at the corners
c         raft "NFTNG"
c
          XA=MIN(XA,CLPR(I,1))
          YA=MIN(YA,CLPR(I,2))
          XD=(MAX(XD,CLPR(I,1)+CLPR(I,3)))
          YD=(MAX(YD,CLPR(I,2)+CLPR(I,4)))
        END IF
 10   CONTINUE
C
c     Determine the co-ordinates of the 4 corners of raft "NFTNG" by
c     subtracting or adding 1 to the co-ordinates of the 4 columns at
c     the corners raft "NFTNG"
c
      XA=XA-1.
      XD=XD+1.
      YA=YA-1.
      YD=YD+1.
C
c     Calculate the plan dimensions and thickness of the raft
c
      XL=XD-XA
      YL=YD-YA
C
      FT=21.*MATRS(ICOL,1)
C
c     Calculate the area of the raft such that the soil pressure
c     at service loads does not exceed allowable bearing capacity
c
 20   AREA=(1+DL4)*TLOAD/ABCAP
c
c     If the calculated area above is greater than the area of the
c     plan dimensions above, adjust the dimensions to match the
c     calculated area
c
 21   IF(AREA.GT.XL*YL)THEN
        F=XL/YL
        YL=SQRT(AREA/F)
        XL=AREA/YL
        ADD=XL-XD+XA
        XA=XA-.5*ADD
        XD=XD+.5*ADD
        ADD=LY-YD+YA
```

```
        YA=YA-.5*ADD
        YD=YD+.5*ADD
        ARE=AREA
      ELSE
        AREA=XL*YL
      END IF
C
c     Assume effective depth of the raft
c
 25   ED=FT-50.-ROD
C
c     Calculate the weight of the raft and check that it is not
c     greater than the one assumed earlier
c
      FTWT=XL*YL*FT*24.E-3
      IF(FTWT*.98.GT.DL4*TLOAD.AND.ARE.GT.XL*YL .OR.FTWT.LT..8*TLOAD .AN
     1D.ARE.GT.XL*YL)THEN
        DL4=FTWT/TLOAD
        GOTO 20
      END IF
C
c     Calculate the co-ordinates of the centre of the raft
c
      XCENT=XA+.5*XL
      YCENT=YA+.5*YL
C
c     Sum moments about the left edge of the raft
c
      RM=0.
      DO 30 I=ICOL,NCOLS
        IF(IST(I).EQ.IST(ICOL))THEN
          RL=CLPR(I,5)+CLPR(I,6)
          R= CLPR(I,1)+CLPR(I,3)*.5
          R1=R-XA
          RM=RM+R1*RL
        END IF
 30   CONTINUE
C
c     Calculate the x-co-ordinate of the point of load resultant
c
      XR=RM/TLOAD+XA
c
c     Calculate eccentricity along x-axis
c
      EX=XR-XCENT
C
c     Calculate eccentricity along y-axis
c
      RM=0.
      DO 35 I=1,NCOLS
       IF(IST(I).EQ.IST(ICOL))THEN
        RL=CLPR(I,5)+CLPR(I,6)
        R= CLPR(I,2)+CLPR(I,4)*.5
        R1=R-YA
        RM=RM+R1*RL
       END IF
 35   CONTINUE
C
      YR=RM/TLOAD+YA
      EY=YR-YCENT
C
      IF(TOREAD .EQ. 'Y')THEN
        WRITE(*,'(A)')' ENTER THE NUMBER OF PANELS ON THIS RAFT'
        READ(*,'(I2)')NPNL
C
```

```
        NPANEL(IRAFT)=NPNL
C
        K=0
        DO 36 I=1,NPNL
         K=K+1
         WRITE(*,'(A,I2)')' ENTER THE 4 COLS AT THE CORNERS OF PANEL',K
         IRAF(I,9)=IRAFT
         READ(*,*)(IRAF(K,L),L=1,4)
  36    CONTINUE

      END IF
C
      VOL=XL*YL*FT*1.E-3
c
c     Initialise the weight of reinforcement
c
      WT=0.
      IPANEL=0
c
c     Taking the panels one by one calculate their reinforcements
c
      DO 60 I=1,NPNL
        IPANEL=IPANEL+1
        SX=9999.
        SY=9999.
        DO 40 M=1,NCOLS
          IF(M.EQ.IRAF(I,1).OR.M.EQ.IRAF(I,2).OR.M.EQ.IRAF(I,3).OR.M.
     1    EQ.IRAF(I,4))THEN
            SX=MIN(SX,CLPR(M,1))
            SY=MIN(SY,CLPR(M,2))
          END IF
 40     CONTINUE
C
        BX=0.
        BY=0.
        DO 41 M=1,NCOLS
          IF(M.EQ.IRAF(I,1).OR.M.EQ.IRAF(I,2).OR.M.EQ.IRAF(I,3).OR.M.
     1    EQ.IRAF(I,4))THEN
            BX=MAX(BX,CLPR(M,1))
            BY=MAX(BY,CLPR(M,2))
          END IF
 41     CONTINUE
C
        N=0
        DO 42 M=ICOL,NCOLS
          N=N+1
          IF(CLPR(M,1).EQ.SX.AND.CLPR(M,2).EQ.SY)IPC(I,1)=N
          IF(CLPR(M,1).EQ.BX.AND.CLPR(M,2).EQ.SY)IPC(I,2)=N
          IF(CLPR(M,1).EQ.SX.AND.CLPR(M,2).EQ.BY)IPC(I,3)=N
          IF(CLPR(M,1).EQ.BX.AND.CLPR(M,2).EQ.BY)IPC(I,4)=N
 42     CONTINUE
C
        IO=IPANEL
        IP=1
        IQ=3
        PXL=((CLPR(IPC(I,4),IP)+.5*CLPR(IPC(I,4),IQ))-(C
     1  LPR(IPC(I,3),IP)+.5*CLPR(IPC(I,3),IQ))+(CLPR(IPC(
     2  I,2),IP)+.5*CLPR(IPC(I,2),IQ))-(CLPR(IPC(I,1),IP)+.5*CLPR(I
     3  PC(I,1),IQ)))*.5

C
        IP=2
        IQ=4
        PYL=((CLPR(IPC(I,4),IP)+.5*CLPR(IPC(I,4),IQ))-(C
     1
```

```
LPR(IPC(I,2),IP)+.5*CLPR(IPC(I,2),IQ))+(CLPR(IPC(
     2  I,3),IP)+.5*CLPR(IPC(I,3),IQ))-(CLPR(IPC(I,1),IP)+.5*CLPR(I
     3  PC(I,1),IQ)))*.5

C
        N=0
        DO 43 M=1,4
          N=N+1
          XCOL=(CLPR(IPC(I,N),1)+.5*CLPR(IPC(I,N),3))
          YCOL=(CLPR(IPC(I,N),2)+.5*CLPR(IPC(I,N),4))
          E=YCOL-YCENT
          IF(E.EQ.0)E=1.
          C=XCOL-XCENT
          IF(C.EQ.0)C=1.
          ulod=1.4*clpr(ipc(i,n),5)+1.6*clpr(ipc(i,n),6)
          are=pxl*pyl
          PWT=FT*PXL*PYL*24.E-3
          IRAF(I,4+N)=INT(((ULOD+1.4*PWT)/ARE+ULOD*EY*E/C+ULOD*EX*C
     1    /E)+.5)
 43     CONTINUE
C
        PAV=((IRAF(I,5)+IRAF(I,6)+IRAF(I,7)+IRAF(I,8)))/4.
        UPN=PAV-FT*33.6E-3
C
        HCX=(CLPR(IPC(I,1),3)+CLPR(IPC(I,2),3)+CLPR(IPC(I,3),3)+CLPR(IPC
     1  (I,4),3))*.17
        HCY=(CLPR(IPC(I,1),4)+CLPR(IPC(I,2),4)+CLPR(IPC(I,3),4)+CLPR(IPC
     1  (I,4),4))*.17
        HC=SQRT(1.27*HCX*HCY)
C
        DMX=.125*UPN*PYL*(PXL-(.67*HC))**2
        DMY=.125*UPN*PXL*(PYL-(.67*HC))**2
        IF(DMX.GT..15*PYL*MATRS(ICOL,2)*ED**2 .OR. DMY.GT..15*PXL*MATRS(
     1    ICOL,2)*ED**2)THEN
          FT=FT*1.1
          TOREAD='N'
          GOTO 25
        END IF
        PMMT(I,1)=DMX
        PMMT(I,6)=DMY
C
        CSMP=.46*DMX
        PMMT(I,2)=CSMP
        CSMN=.22*DMX
        PMMT(I,3)=CSMN
C
        SMP=.16*DMX
        PMMT(I,4)=SMP
        SMN=.16*DMX
        PMMT(I,5)=SMN
C
        CSMP=.46*DMY
        PMMT(I,7)=CSMP
        CSMN=.22*DMY
        PMMT(I,8)=CSMN
C
        SMP=.16*DMY
        PMMT(I,9)=SMP
        SMN=.16*DMY
        PMMT(I,10)=SMN
C
C Calculate As per m width
c
```

```
PRENF(I,1)=PMMT(I,2)*1.E6/(.83*MATRS(ICOL,3)*ED)
        SW=.5*PXL
        IF(PRENF(I,1) .LT. 1.5*SW*ED)PRENF(I,1)=1.5*SW*ED
        ASPM=PRENF(I,1)/SW
        write(*,'(a,f9.2)')' prenf(i,1)',prenf(i,1)
        READ(*,'(A)')ANS
C
        IB=1
 45     BR=BARS(IB)
c
c Calculate nr. of bars per m width
c
        IRX(I,1)=INT(ASPM/(.79*BR**2)+.5)
c
c Calculate spacing of bars  per m width
c
        IRX(I,3)=1.E3/IRX(I,1)
        IF(IRX(I,3) .LT. 125)then
          IB=IB+1
          IF(IB .GT. 5)THEN
            FT=FT*1.2
            TOREAD='N'
            GOTO 25
          END IF
          GOTO 45
        ELSE IF(IRX(I,3) .GT. 300)THEN
          IRX(I,3)=300
        END IF
        IRX(I,1)=INT(ASPM/IRX(I,3)+.5)
        ASPM=IRX(I,1)*BR**2*.79/SW
        PRENF(I,1)=IRX(I,1)*BR**2*.79
        IRX(I,2)=BR
        IRX(I,4) = PRENF(I,1)*100./(ED*1.E3)
        CAR=22.*BR**2/28.
        WT=WT+.00786*(CAR*FLOAT(IRX(I,1)))*PYL*.5
C
        PRENF(I,2)=PMMT(I,3)*1.E6/(.83*MATRS(ICOL,3)*ED)
        SW=.5*PXL
        IF(PRENF(I,2) .LT. 1.5*SW*ED)PRENF(I,2)=1.5*SW*ED
        ASPM=PRENF(I,2)/SW
C
        IB=1
 46     BR=BARS(IB)
C
        IRX(I,5)=INT(ASPM/(.79*BR**2)+.5)
C
        IRX(I,7)=1.E3/IRX(I,5)
        IF(IRX(I,7) .LT. 125)then
          IB=IB+1
          IF(IB .GT. 5)THEN
            FT=FT*1.2
            TOREAD='N'
            GOTO 25
          END IF
          GOTO 46
        ELSE IF(IRX(I,5) .GT. 300)THEN
          IRX(I,5)=300
        END IF
        IRX(I,5)=INT(ASPM/IRX(I,7)+.5)
        ASPM=IRX(I,5)*BR**2*.79/SW
        PRENF(I,2)=IRX(I,5)*BR**2*.79
        IRX(I,6)=BR
        IRX(I,8) = PRENF(I,2)*100./(ED*1.E3)
        CAR=22.*BR**2/28.
```

```
      WT=WT+.00786*(CAR*FLOAT(IRX(I,5)))*PYL*.5
C
            PRENF(I,3)=PMMT(I,4)*1.E6/(.83*MATRS(ICOL,3)*ED)
            SW=.5*PXL
            IF(PRENF(I,3) .LT. 1.5*SW*ED)PRENF(I,3)=1.5*SW*ED
            ASPM=PRENF(I,3)/SW
C
            IB=1
 47         BR=BARS(IB)
C
            IRX(I,9)=INT(ASPM/(.79*BR**2)+.5)
C
            IRX(I,11)=1.E3/IRX(I,9)
            IF(IRX(I,11) .LT. 125)then
              IB=IB+1
              IF(IB .GT. 5)THEN
                FT=FT*1.2
                TOREAD='N'
                GOTO 25
              END IF
              GOTO 47
            ELSE IF(IRX(I,9) .GT. 300)THEN
              IRX(I,9)=300
            END IF
            IRX(I,9)=INT(ASPM/IRX(I,11)+.5)
            ASPM=IRX(I,9)*BR**2*.79/SW
            PRENF(I,3)=IRX(I,9)*BR**2*.79
            IRX(I,10)=BR
            IRX(I,12) = PRENF(I,3)*100./(ED*1.E3)
            CAR=22.*BR**2/28.
            WT=WT+.00786*(CAR*FLOAT(IRX(I,9)))*PYL*.5
C
            PRENF(I,4)=PMMT(I,5)*1.E6/(.83*MATRS(ICOL,3)*ED)
            SW=.5*PXL
            IF(PRENF(I,4) .LT. 1.5*SW*ED)PRENF(I,4)=1.5*SW*ED
            ASPM=PRENF(I,4)/SW
C
            IB=1
 48         BR=BARS(IB)
C
            IRX(I,13)=INT(ASPM/(.79*BR**2)+.5)
C
            IRX(I,15)=1.E3/IRX(I,13)
            IF(IRX(I,15) .LT. 125)then
              IB=IB+1
              IF(IB .GT. 5)THEN
                FT=FT*1.2
                TOREAD='N'
                GOTO 25
              END IF
              GOTO 48
            ELSE IF(IRX(I,13) .GT. 300)THEN
              IRX(I,13)=300
            END IF
            IRX(I,13)=INT(ASPM/IRX(I,15)+.5)
            ASPM=IRX(I,13)*BR**2*.79/SW
            PRENF(I,4)=IRX(I,13)*BR**2*.79
            IRX(I,14)=BR
            IRX(I,16) = PRENF(I,4)*100./(ED*1.E3)
            CAR=22.*BR**2/28.
            WT=WT+.00786*(CAR*FLOAT(IRX(I,13)))*PYL*.5
C
            PRENF(I,5)=PMMT(I,7)*1.E6/(.83*MATRS(ICOL,3)*ED)
            SW=.5*PYL
            IF(PRENF(I,5) .LT.        1.5*SW*ED)PRENF(I,5)=1.5*SW*ED
```

```
      ASPM=PRENF(I,5)/SW
C
      IB=1
 49   BR=BARS(IB)
C
      IRX(I,17)=INT(ASPM/(.79*BR**2)+.5)
C
      IRX(I,19)=1.E3/IRX(I,17)
      IF(IRX(I,19) .LT. 125)then
        IB=IB+1
        IF(IB .GT. 5)THEN
          FT=FT*1.2
          TOREAD='N'
          GOTO 25
        END IF
        GOTO 49
      ELSE IF(IRX(I,17) .GT. 300)THEN
        IRX(I,17)=300
      END IF
      IRX(I,17)=INT(ASPM/IRX(I,19)+.5)
      ASPM=IRX(I,17)*BR**2*.79/SW
      PRENF(I,5)=IRX(I,17)*BR**2*.79
      IRX(I,18)=BR
      IRX(I,20) = PRENF(I,5)*100./(ED*1.E3)
      CAR=22.*BR**2/28.
      WT=WT+.00786*(CAR*FLOAT(IRX(I,17)))*PXL*.5
C
      PRENF(I,6)=PMMT(I,8)*1.E6/(.83*MATRS(ICOL,3)*ED)
      SW=.5*PYL
      IF(PRENF(I,6) .LT. 1.5*SW*ED)PRENF(I,6)=1.5*SW*ED
      ASPM=PRENF(I,6)/SW
C
      IB=1
 50   BR=BARS(IB)
C
      IRX(I,21)=INT(ASPM/(.79*BR**2)+.5)
C
      IRX(I,23)=1.E3/IRX(I,21)
      IF(IRX(I,23) .LT. 125)then
        IB=IB+1
        IF(IB .GT. 5)THEN
          FT=FT*1.2
          TOREAD='N'
          GOTO 25
        END IF
        GOTO 50
      ELSE IF(IRX(I,21) .GT. 300)THEN
        IRX(I,21)=300
      END IF
      IRX(I,21)=INT(ASPM/IRX(I,23)+.5)
      ASPM=IRX(I,21)*BR**2*.79/SW
      PRENF(I,6)=IRX(I,21)*BR**2*.79
      IRX(I,22)=BR
      IRX(I,24) = PRENF(I,6)*100./(ED*1.E3)
      CAR=22.*BR**2/28.
      WT=WT+.00786*(CAR*FLOAT(IRX(I,21)))*PXL*.5
C
      PRENF(I,7)=PMMT(I,9)*1.E6/(.83*MATRS(ICOL,3)*ED)
      SW=.5*PYL
      IF(PRENF(I,7) .LT. 1.5*SW*ED)PRENF(I,7)=1.5*SW*ED
      ASPM=PRENF(I,7)/SW
C
      IB=1
 51   BR=BARS(IB)
C
```

```
      IRX(I,25)=INT(ASPM/(.79*BR**2)+.5)
C
      IRX(I,27)=1.E3/IRX(I,25)
      IF(IRX(I,27) .LT. 125)then
        IB=IB+1
        IF(IB .GT. 5)THEN
          FT=FT*1.2
          TOREAD='N'
          GOTO 25
        END IF
        GOTO 51
      ELSE IF(IRX(I,25) .GT. 300)THEN
        IRX(I,25)=300
      END IF
      IRX(I,25)=INT(ASPM/IRX(I,27)+.5)
      ASPM=IRX(I,25)*BR**2*.79/SW
      PRENF(I,7)=IRX(I,25)*BR**2*.79
      IRX(I,26)=BR
      IRX(I,28) = PRENF(I,7)*100./(ED*1.E3)
      CAR=22.*BR**2/28.
      WT=WT+.00786*(CAR*FLOAT(IRX(I,25)))*PXL*.5
C
      PRENF(I,8)=PMMT(I,10)*1.E6/(.83*MATRS(ICOL,3)*ED)
      SW=.5*PYL
      IF(PRENF(I,8) .LT. 1.5*SW*ED)PRENF(I,8)=1.5*SW*ED
      ASPM=PRENF(I,8)/SW
C
      IB=1
 52   BR=BARS(IB)
C
      IRX(I,29)=INT(ASPM/(.79*BR**2)+.5)
C
      IRX(I,31)=1.E3/IRX(I,29)
      IF(IRX(I,31) .LT. 125)then
        IB=IB+1
        IF(IB .GT. 5)THEN
          FT=FT*1.2
          TOREAD='N'
          GOTO 25
        END IF
        GOTO 52
      ELSE IF(IRX(I,29) .GT. 300)THEN
        IRX(I,29)=300
      END IF
      IRX(I,29)=INT(ASPM/IRX(I,31)+.5)
      ASPM=IRX(I,29)*BR**2*.79/SW
      PRENF(I,8)=IRX(I,29)*BR**2*.79
      IRX(I,30)=BR
      IRX(I,34) = PRENF(I,8)*100./(ED*1.E3)
      CAR=22.*BR**2/28.
      WT=WT+.00786*(CAR*FLOAT(IRX(I,29)))*PXL*.5
C
C  CHECK AVE. OF BAR DIAMETERS USED IN CALCULATING ED
C
      IF(.5*(IRX(I,2)+IRX(I,18)).GT.ROD)THEN
        ROD=.5*(IRX(I,2)+IRX(I,18))
        TOREAD='N'
        GOTO 25
      END IF
C
      IF(.5*(IRX(I,6)+IRX(I,22)).GT.ROD)THEN
        ROD=.5*(IRX(I,6)+IRX(I,22))
        TOREAD='N'
        GOTO 25
      END IF
```

```
C
      IF(.5*(IRX(I,10)+IRX(I,26)).GT.ROD)THEN
        ROD=.5*(IRX(I,10)+IRX(I,26))
        TOREAD='N'
        GOTO 25
      END IF
C
      IF(.5*(IRX(I,14)+IRX(I,30)).GT.ROD)THEN
        ROD=.5*(IRX(I,14)+IRX(I,30))
        TOREAD='N'
        GOTO 25
      END IF
C
      XLC=MIN(CLPR(IPC(I,1),3),CLPR(IPC(I,2),3),CLPR(IPC(I,3),3),CLPR(
     1  IPC(I,4),3))
      YLC=MIN(CLPR(IPC(I,1),4),CLPR(IPC(I,2),4),CLPR(IPC(I,3),4),CLPR(
     1  IPC(I,4),4))
      CP=4.*MIN(XLC*1.E3,YLC*1.E3)+9.43*FT
      AWP=(MIN(XLC*1.E3,YLC*1.E3)+3.*FT)**2-(.86*(1.5*FT)**2)
      PSF=UPN*(PXL*PYL-(AWP*1.E-6))
C
      IF(PSF .LT. 0.)PSF=0.
      PSS=PSF*1.E3/CP/ED
C
      FD=400./ED
      IF(FD .LT. 1.)FD=1.
      IF(MATRS(ICOL,2) .LE. 25)RK2=1.
      IF(MATRS(ICOL,2) .GT. 25 .AND. MATRS(ICOL,2) .LE. 40)RK2=(MATRS(
     1  ICOL,2)/25)**.33
      ROX=PRENF(I,1)/(.5*PXL)/ED*1.E-1
      ROY=PRENF(I,5)/(.5*PYL)/ED*1.E-1
C
 53   RO=0.5*(ROX+ROY)
C
      USSX=RK2*.79*RO**.33*FD**.25
      USSY=RK2*.79*RO**.33*FD**.25
C
      IF(PSS .GT. MIN(USSX,USSY))THEN
        if(ro .lt. 1.5)then
          WT=WT-.00786*(CAR*FLOAT(IRX(I,1)))*PYL*.5
          rox=rox*1.2
          SW=.5*PXL
          PRENF(I,1)=5*ROX*PXL*ED
          ASPM=PRENF(I,1)/SW
C
          IB=1
 54       BR=BARS(IB)
C
          IRX(I,1)=INT(ASPM/(.79*BR**2)+.5)
C
          IRX(I,3)=1.E3/IRX(I,1)
          IF(IRX(I,3) .LT. 125)then
           IB=IB+1
           IF(IB .GT. 5)THEN
             FT=FT*1.2
             TOREAD='N'
             GOTO 25
           END IF
           GOTO 54
          ELSE IF(IRX(I,3) .GT. 300)THEN
           IRX(I,3)=300
          END IF
          IRX(I,1)=INT(ASPM/IRX(I,3)+.5)
          ASPM=IRX(I,1)*BR**2*.79/SW
                                PRENF(I,1)=IRX(I,1)*BR**2*.79
```

```
            IRX(I,2)=BR
            IRX(I,4) = PRENF(I,1)*100./(ED*1.E3)
            CAR=22.*BR**2/28.
            WT=WT+.00786*(CAR*FLOAT(IRX(I,1)))*PYL*.5
            roy=roy*1.2
            WT=WT-.00786*(CAR*FLOAT(IRX(I,17)))*PXL*.5
            SW=.5*PYL
            PRENF(I,5)=5*ROY*PYL*ED
            IF(PRENF(I,5) .LT. 1.5*SW*ED)PRENF(I,5)=1.5*SW*ED
            ASPM=PRENF(I,5)/SW
C
            IB=1
 55         BR=BARS(IB)
C
            IRX(I,17)=INT(ASPM/(.79*BR**2)+.5)
C
            IRX(I,19)=1.E3/IRX(I,17)
            IF(IRX(I,19) .LT. 125)then
             IB=IB+1
             IF(IB .GT. 5)THEN
               FT=FT*1.2
               TOREAD='N'
               GOTO 25
             END IF
             GOTO 55
            ELSE IF(IRX(I,17) .GT. 300)THEN
             IRX(I,17)=300
            END IF
             IRX(I,17)=INT(ASPM/IRX(I,19)+.5)
             ASPM=IRX(I,17)*BR**2*.79/SW
             PRENF(I,5)=IRX(I,17)*BR**2*.79
             IRX(I,18)=BR
             IRX(I,20) = PRENF(I,5)*100./(ED*1.E3)
             CAR=22.*BR**2/28.
             WT=WT+.00786*(CAR*FLOAT(IRX(I,17)))*PXL*.5
             goto 53
           end if
           FT=FT*1.1
           TOREAD='N'
           GOTO 25
         END IF
 60    CONTINUE
C
            AR(NFTNG,1)=FT
            AR(NFTNG,2)=XL
            AR(NFTNG,3)=YL
            AR(NFTNG,4)=XA
            AR(NFTNG,5)=XD
            AR(NFTNG,6)=YA
            AR(NFTNG,7)=YD
            AR(NFTNG,8)=VOL
            AR(NFTNG,9)=WT
            AR(NFTNG,10)=WT/VOL
            RETURN
            END
C
C
C
            SUBROUTINE HELP
            CHARACTER ANS*3
C
            WRITE(*,10)' ABOUT HELP',' The following topics are covered:',
           1' 1  About the program',
           2' 2  Input required',
```

```
     3' 3  How to enter input from the screen',
     4' 4  How to enter input from a data file',
     5' 5  Example  on how to enter input from a data file',
     6' 6  How to choose type of footing',
     7' 7  Tips',
     8' The topics appear in the order in which they are listed.',
     9' To skip a topic type "NO"; to view the topic hit return key.'
 10   FORMAT(//a//a/a/a/a/a/a/a/a//a/a//)
 11   FORMAT(a//a/a/a/a//a/a//a/a//a/)
C
      WRITE(*,'(//A/)')' ABOUT THE PROGRAM?'
      READ(*,'(A)')ANS
      IF(ANS .NE. 'NO')THEN
        WRITE(*,11)' This program designs the following footings: ',
     1' 1 Single pad footings',
     2' 2 Combined footings',
     3' 3 Strip footings',
     4' 4 Raft foundations.',
     5' If none of the footings above is a solution, the program',
     6' suggests a deep foundation.',
     7' The program is user friendly and all questions  must be',
     8' answered in uppercase letters.',
     9' Results are stored in file OUT. To see them edit the file'
C
      WRITE(*,'(/A)')' Hit return key to continue'
      READ(*,'(A)') ANS
C
      WRITE(*,9)' PROGRAM LIMITATIONS',
     1' The program is limited by the following:',
     2'   Only axial loads and rectangular columns are considered',
     3'   Columns must be arranged in an orthogonal grid using',
     4'   Cartesian rectangular coordinates',
     5'   A maximum of 99 columns only can be considered at a time',
     6'   Data file name must not exceed 8 characters'
      END IF
  9   FORMAT(/a//a//a/a/a/a/a)
C
      WRITE(*,'(//A)')' INPUT REQUIRED? '
      READ(*,'(A)')ANS
      IF(ANS .NE. 'NO')THEN
        WRITE(*,12)' The main input required are:',
     1' 1 Data on column locations and loads ',
     2' 2 Types of foundation to be designed.',
     3' Input on column locations is (i.e taking col section on plan):',
     4' 1 x-coordinate of lower left corner',
     5' 2 y-coordinate of lower left corner',
     6' 3 dimension along x-axis in mm',
     7' 4 dimension along y-axis in mm',
     8' Input on column loads is',
     9' dead and live loads'
 12   FORMAT(a//a/a//a//a/a/a/a/a//a,a/)
C
      WRITE(*,'(/A)')' Hit return key to continue'
      READ(*,'(A)') ANS
C
      WRITE(*,'(a/a/a)')' The input can be entered either from the',
     1' screen or from a data file. When asked which option is',
     2' chosen, enter "S" if it is screen; "F" if it is data file'
      WRITE(*,'(/A/A//A/A)')' In addition the following input are also',
     1' required for RAFTS:',
     2' 1 Nr. of panels on the raft',
     3' 2 The columns at the  corners of each panel'
      END IF
C
      WRITE(*,'(//A)')' HOW TO    ENTER INPUT FROM SCREEN?'
```

```
      READ(*,'(A)')ANS
      IF(ANS .NE. 'NO')THEN
       WRITE(*,13)' If data is entered from the screen, the required',
     1' input will be asked for. Just enter the input at the prompt.'
      END IF
 13   FORMAT(/a/a)
C
      WRITE(*,'(//A)')' HOW TO ENTER INPUT FROM DATA FILE?'
      READ(*,'(A)')ANS
      IF(ANS .NE. 'NO')THEN
       WRITE(*,14)' To use this option you must have stored  all data',
     1' in a file. When asked for the name of the file enter it.',
     2' To store data in your input file follow the format below:',
     3' 1  NCOLS',
     4' (the next pair of lines- 2&3- are entered as many times as the',
     5' number of columns)',
     6' 2  X Y Lx Ly DL LL',
     7' 3  DB ',
     8' 4  ABCAP',
     9' 5  Fcu Fy'
 14   FORMAT(//a/a/a/a/a/a/a/a/a/a)
C
      WRITE(*,'(/A)')' Hit return key to continue'
      READ(*,'(A)') ANS
C
        WRITE(*,15)' The symbols above are:',
     1' NCOLS: No. of columns',
     2' X:     x-coordinate of lower left corner',
     3' Y:     y-coordinate of lower left corner',
     4' Lx;    dimension along x-axis in mm',
     5' Ly:    dimension along y-axis in mm',
     6' DL:    Dead load in kN',
     7' LL:    Live load in kN',
     8' DB:    Dowel bar size in mm',
     9' ABCAP: Allowable bearing capacity in kN/m2'
        WRITE(*,'(a)')' Fcu:   Concrete cube strength in N/mm2',' Fy:

     1Steel strength in N/mm'
      END IF
 15   FORMAT(/a//a/a/a/a/a/a/a/a/a)
C
      WRITE(*,'(//A)')' EXAMPLE ON HOW TO ENTER INPUT FROM DATA FILE?'
      READ(*,'(A)')ANS
      IF(ANS .NE. 'NO')THEN
        WRITE(*,16)' Given the following column information:',
     1' No. of columns = 2',
     2' Col 1: X  = 15',
     3'        Y  = 17.8',
     4'        Lx = 300',
     5'        Ly = 300',
     6'        DL = 1000',
     7'        LL = 200',
     8'        DB = 25'
C
      WRITE(*,17)' Col 2: X  = 17.95',
     1'        Y  = 17.75',
     2'        Lx = 400',
     3'        Ly = 400',
     4'        DL = 1400',
     5'        LL = 300',
     6'        DB = 25'
C
      WRITE(*,'(a)')' ABCAP=300',' Fcu = 30',' Fy = 425'
      WRITE(*,'(/a)')' Hit return key to continue'
      READ(*,'(a)')ans
```

```
      WRITE(*,'(/a)')' Your input file will look like this:'
      WRITE(*,18)' 2',
     1' 15 17.8 300 300 1000 200',
     2' 25',
     3' 17.95 17.75 400 400 1400 300',
     4' 25',
     5' 300',
     6' 30 425'
      END IF
 16   FORMAT(/a//a/a/a/a/a/a/a/a)
 17   FORMAT(a/a/a/a/a/a/a)
 18   FORMAT(/a/a/a/a/a/a/a)
C
      WRITE(*,'(//A)')' HOW TO CHOOSE FOOTING TYPES?'
      READ(*,'(A)')ANS
      IF(ANS .NE. 'NO')THEN
       WRITE(*,'(a)')' Choosing footing types is done interactively with
     1 the computer. Just answer the questions you are asked'
      END IF
C
      WRITE(*,'(//A)')' TIPS?'
      READ(*,'(A)')ANS
      IF(ANS .NE. 'NO')THEN
       WRITE(*,19)' 1 Set out your data as it would appear in the data',
     1'    file for easy reference',
     2' 2 Note that most times you will have coordinates of column',
     3'    center. What you need are coordinates of LOWER LEFT CORNER!!',
     4'    To get them, subtract half the column dimension from the',
     5'    corresponding center coordinate'
 19   FORMAT(/a/a//a/a/a/a)
      WRITE(*,20)' 3 If you have many columns or groups of columns',
     1'    that have the same loading and footing type design for one',
     2'    and use the result for the others. This way you only have to',
     3'    enter data for one column in case of single footings or data',
     4'    for one set of columns in case of the other footing types'
      WRITE(*,'(/a)')' More tips? '
      READ(*,'(a)')ANS
      IF(ANS .EQ. 'NO') GOTO 30
      WRITE(*,21)' 4 You can play around with your results by',
     1'    changing some data in your input file e.g  bearing capacity',
     2'    cube strength etc. This will show you how they affect',
     3'    your output'
      WRITE(*,22)' 5 All questions have default values. To use',
     1'    default values press return key. For default values see',
     2'    program  manual'
      WRITE(*,'(/a)')' 6 All answers must be in uppercase letters!!'
      WRITE(*,'(/a/a)')' 7 If you are designing a raft prepare a  ',
     1'  table showing nr. of panels and columns at panel corners'
      END IF
 20   FORMAT(/a/a/a/a/a/a)
 21   FORMAT(/a/a/a/a/a)
 22   FORMAT(a/a/a)
C
 30   CONTINUE
      RETURN
      END
```