RICE UNIVERSITY

TWO EXAMPLES OF COMPUTER APPLICATIONS
IN THE ARCHITECTURAL FIELD

ENRIQUE HERNANDEZ - JAIME

A THESIS SUBMITTED
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF

MASTER IN ARCHITECTURE

Thesis Director

Nat W. Kroh

Houston, Texas

May 1970
ABSTRACT

TWO EXAMPLES OF COMPUTER APPLICATIONS IN THE ARCHITECTURAL FIELD

ENRIQUE HERNANDEZ - JAIME

Computers not only extend our capability for handling large amounts of data and perform sophisticated processes, they also extend our capabilities for making better decisions and exploring widely and deeply different alternatives that otherwise could only be mentioned in passing.

Computers have had a great impact in all kinds of human activities, yet for the most part architects were, and some are still reluctant to their application in this field.

It is argued that computers can not solve architectural problems, owing to the great subjectiveness involved, or that they tend to obliterate creativity.

It is intended here to demonstrate, in the first case, that the computer is not used to "solve" the problem but to help to analyze and make better decisions in a more rational and systematic way and, in the second case that this help will broaden architect's creativity by helping him respond realistically to the problems and by increasing his means to test the validity of his solutions.

It is assumed that it is still, and will always be, up to
the architect to be creative using computers, and not the reverse. It is only through a systematic approach that problems will be stated properly and that solutions can be expected to approach or reach optimality.

Intuition, experience and creativity must be present throughout the whole process of architecture, complementing rather than supplementing rational decisions.

Computers do not provide merely numbers, but most importantly the means for evaluating the consequences of making decisions, before they are implemented.

Part I of this work is a brief introduction to the ideas leading to the author's arguments, and exposes in detail two cases studied, one for data processing and one for problem solving. Documents complementing this discussion are presented for reference as appendices.

Part II deals with some references about experiences already obtained in operations research work, continuing with part III which deals with the relevance of computers in the architectural process, and the importance of interfacing devices (Input/Output).

Part IV is a brief summary and statement of conclusions and finally part V presents appendices and a bibliography.
TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>PART</th>
<th>SECTION</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>TWO CASES OF STUDY</td>
<td>4</td>
</tr>
<tr>
<td>II</td>
<td>EXPERIENCE IN THE FIELD OF OPERATIONS RESEARCH,</td>
<td>92</td>
</tr>
<tr>
<td></td>
<td>CONSIDERED FOR ARCHITECTURE</td>
<td></td>
</tr>
<tr>
<td>III</td>
<td>RELEVANCE OF COMPUTERS IN PROBLEM DEFINITION,</td>
<td>99</td>
</tr>
<tr>
<td></td>
<td>PROBLEM SOLVING AND SOLUTION IMPLEMENTATION.</td>
<td></td>
</tr>
<tr>
<td>IV</td>
<td>SUMMARY AND CONCLUSIONS</td>
<td>111</td>
</tr>
<tr>
<td>V</td>
<td>APPENDICES (FLOWCHARTS, SOURCE PROGRAMS AND</td>
<td>115</td>
</tr>
<tr>
<td></td>
<td>SAMPLES OF INPUT/OUTPUT), BIBLIOGRAPHY.</td>
<td></td>
</tr>
</tbody>
</table>
INTRODUCTION

Computer systems have become the most important tool in any kind of problem solving process, not only by handling and processing enormous amounts of data, or even by carrying out the most complex and sophisticated computational procedures, but also by enabling man to make use of information, when it is needed, and by inducing the development of more sophisticated techniques of analysis, for more rational and effective ways for applying them.

The computer itself is not relevant, what is relevant is the use we make of it, and for that we need to know what it can do and what it can not do, what can we expect, through its applications, and how can we make use of it.

"It is impossible to think and talk usefully about computers if there is not at least a rudimentary knowledge of this technology"\(^1\)

However computers can not be used simply because "they are here", there must be the need for their application, and in fact the need is already present, and many architectural firms are already using them in a wide variety of applications at several levels of complexity.

Problems grow rapidly in complexity, and new kins of problems, not known before, have to be approached; enormous amounts

---

of information have to be handled, analyzed and processed; demands of more efficient and rational allocation of resources and time as a crucial factor oblige to more rational decision making; more sophisticated clients want to be involved in this process, all theses making it impossible for one man's capabilities to cope with the whole problem.

Definite trends towards multidisciplinary team approach to architectural problems are in effect, in order to apply more systematic methods for finding and solving the problems and implementing that solutions.

Architects are no more inspired form giver-artists, but rather members of such teams who have to work within those teams and must learn how to do it better, joining their efforts to those of people from other fields in the whole process in order to come out with realistic solutions, to real not imagined problems.

These brief introductory ideas lead to the statement of the author's thesis:

It is his aim not to overemphazise the role played by a computer in this process, but to demonstrate that it is the most powerful tool that team members ever had and that, when properly understood and applied, in an efficient man-machine process, it largely extends their capability for better decision making throughout the entire project
delivery system and implementation.\textsuperscript{2}

And finally, this computer-team approach not only does not preclude intuition and creativity, but perhaps will also lead to new and more responsible formal expressions of our computerized age.

The work has been divided in five parts.

Part I deals with the brief introduction given above, and two cases of study worked by the author, which are essentially the body of the work. They are presented as examples of computer applications, and do not intend to cover complete processes, which is a task to be undertaken by a team, but do intend to show how these processes can be conducted and to prove also his present and next arguments. These two cases of study deal with a data processing example and a structural problem example, and include listing of the programs, documentation and samples of output.

Part II deals with references about operations research experiences leading to part III.

Part III deals with the relevance of computers in problem definition, problem solving and implementation of solutions. These, when added to the assumptions made in the statement of the thesis, and the experience obtained through the cases

\textsuperscript{2} W. Pena. Problem Seeking. 1969.
of study, are intended to demonstrate the validity of the arguments.

Part IV deals with a summary and brief statements of the author's conclusions.

Part V contains appendices with program listings, flow charts, samples of output, and bibliography.

I.1. Cases of study.

Two cases of study are presented in this work, which comprise a set of programs developed by the author as part of his training in the field of computers, in order to substantiate the concepts stated here, and essentially as experiences which support further development in his future professional activities.

The first case is developed in order to analyze information concerning activities of students as individuals and time allocation. Information is supplied to the computer, concerning where and how they spend their time.

Information is analyzed and summarized by the programs, and displayed as tables and graphs.

Vast amounts of data are processed with a relatively simple arithmetic in most of them, but making emphasis in the logic of the processes.

Data provided by the students was collected during a one week period, twenty-four hours a day, by means of a diary in
which information was recorded.

The case is presented in two different stages:

A) - Duke University
B) - Rice University

Example A involves:

A.1. Tape validation.
A.2. Programs validation
A.2.1. Records adjustment
A.2.2. Man hours distribution.

Example B involves:

B.1. Characteristics and collection of data.
B.4. Man hours distribution. (I)
B.5. Preliminary statistical analysis.

3. Both cases were handled by Rice's Computer System.

4. Records from here on refer to major items in a file, considering a file as a 'package of data', that is stored and processed. In this part a record particularly refers to all items of information relating the performance of one activity in a continued period of time. For a detailed discussion see Stark and McCracken.
The second case of study is developed in order to analyze a simplified structural problem, namely, a system of steel joists and beams. This case analyzes the structural elements for a set of bay sizes ranging from twelve by twelve feet up to forty by forty feet in regular increments of length of two feet. It also considers two different situations:

a) - When beams span length and joists span width
b) - When beams span width and joists span length

The program is able to handle different alternatives, some of which will be presented here.

Data is classified in two categories:

i) - Variable input, referring to data which govern the course of action and conditions to satisfy the particular application.

ii) - Non variable input, referring to data which remain the same in a set of applications, or when studying different alternatives. These data are not definitely fixed in a strict sense, but are less likely to vary.

Computations are more complex though logic is not less important; in fact, changes in process for different alternatives depend more on logical processes than on computational modifications.
In this case different ways of displaying graphical information were explored.

The two case studies are different each other. It is the aim of the author to show two important points, which in fact, are the core of his concepts stated in this thesis:

A) - Computers are not restricted to particular kinds of problems, but rather they can handle any kind of problem challenging our contemporary practice, provided we as architects (not as programmers) have sufficient knowledge about how to communicate and use optimally actual and coming features which form the "computer system".

B) - Programs developed for computer application are not "closed-in" entities, not even finished ones. They have to be viewed as subsystems interacting with each other, and as parts of more complex and sophisticated systems, that comprise the widest range of architect's needs. Furthermore, they are to be modified, updated and even corrected for users, so as to obtain optimum responses, based upon reviews and/or applications.

Development of such comprehensive systems by teams composed not only by architects, is a strongly needed accomplishment, in order to endow these or other teams, with tools for better decisions.

Documentation. It is assumed that proper documentation is at
least as important as the programs themselves; improper documentation or lack of it, make programs obsolete after a short period of time, even for the programer who developed them.

A brief introduction to the subject is provided here in order to make clear the steps followed with that purpose.

Documentation, as intended in this work, deals with four objectives:

1)- As a means of preparation of the program itself.
2)- As a means of checking the process and correcting errors ("debugging")
3)- As a means of modification, extension, validation and updating of programs.
4)- As a means of communication with users other than the programmer, and to facilitate the use of programs.

Four steps are considered in order to accomplish the objectives:

1)- Comment cards, imbedded throughout the source program, as prologue and where a particular note to the process is needed.

5. More detailed information:
   A)- Documentation standards, Max Gray & Keith R. London.
   B)- Computer methods in Civil Engineering, Steven J. Fenves.
Comments on punched cards must be brief and clear and their position must be properly planned, otherwise, they will confuse more than help.

This particular form of documentation is of the most common, and particularly useful at "debugging" and testing stages.

2) - Variable names are intended as much as possible to resemble the real meaning it represents in common language, the purpose being that in this way a natural relation between the program and the real problem represented is developed, making it easier for users to check the source program listing in future uses.

3) - Flow Diagrams: This form of documentation is intended to provide a clear graphical representation of the logical process involved in the programs.

As intended in this work, it is presented in two forms:

Macro-flow Diagrams, showing the main blocks of processing and the main tests (branches) in a unique chart, which intends to provide information about the main features of the problem's logical process.

Micro-flow Diagrams properly cross-referenced with the corresponding macro charts, showing a complete breakdown of the steps of the program, which intends to provide information about the detailed logical process involved in the program.
Particular interest is placed upon these diagrams since they are the principal means for logic development and further preparation of the source program.

4)- Finally, in order to complement rather than supplement documentation forms described above, the user's manual is a detailed and comprehensive document attached to the program itself.

The user's manual has special importance, particularly for users other than the programmer, and in general as intended in this work, it includes the following points:

A) - Description of the function of the program.
B) - Solution method and special procedures.
C) - Assumptions made.
D) - Limitations of the program.
E) - User's options.
F) - Characteristics and formats for input.
G) - Characteristics and formats for output.
H) - Description of the system used.
I) - I/O designators.
J) - Error messages.
K) - Definition of arguments in subprograms.

For examples on comments, lists (object programs) and flow

---

6. Since all programs for both cases were developed for the same system reader is referred to: time sharing system, terminal user's guide and operation manual 1024926.
diagrams, the reader is referred to appendices.

User's manuals will be developed below. They follow, for each example, the points just defined above according to each particular situation.

DUKE VALIDATED PROGRAMS AND VALIDATED INFORMATION

Data preparation, step 1.
Data contained in magnetic tape, developed by CRS and EFL, with the following characteristics:
9 tracks
800 BPI
Record length = 25 characters.
Block size = 7200 characters.

Such tape, applicable to an IBM 360/ system, has to be transferred into another magnetic tape of the B-5500 system, containing essentially the same information as the original tape, but changing its characteristics to:
7 tracks
800 BPI
Record length = 900 words (900 X 8 = 7200)
Block size = 1 record.
Unlabeled; Alpha.

Data preparation, step 2.

A)- Name of the program : EHJ/DUKE1.
Language : Fortran.
Estimated time: 1 hour, 40 minutes.
Compilation time: 11 seconds.
Estimated core storage: 8960 words.

Program used for reading data from validated tape #1, store information of each block in core memory as a series of one dimensional arrays, and by means of comparisons add to each group of data, the activities' ending time, and sequential numbers.

Other variables are added, only as space in the record length, to be used later on.

Output is to tape #2, Label: DUKE2, with modified records and block size, and containing new information about time and space such that: information is completed and the tape organized in an appropriate form, for next steps.

B)- Solution method and procedures. Data contained in tape #1, is read by complete blocks, of 1 record and stored in temporary arrays in core memory,
TEMP1 stores ID
  .
  .
  .
TEMP9 stores SPACE
which is the form in which information is entered. Any time a record is read, several groups of data are sorted in those arrays for process, check about equality of days, and of sample numbers in order to determine ending time of each
of each activity which is then added to the group, together with an activity sequential number.

For each change of day or sample, a new group is created and placed on tape, so that complete information is obtained for each day (24 hours/day), and each sample change.

When data stored in core memory is exhausted, the process is repeated, until the data stored on tape #1 has been also exhausted.

C)- Assumptions made. Information about hours is given to the nearest sixth of an hour (10 minutes).

D)- Limitations. Program is limited to the case of handling simultaneously magnetic tapes for two different systems.

F)- Characteristics and formats for input:

7 tracks.

800 BPI

Record length = 900 words (8 characters each)

Block size = 1 record.

Unlabeled, and alphanumeric.

Magnetic tape used as input, is organized in such a way that a record of 900 words constitutes a complete item to be handled by the computer at a time, every time a record is read from tape, 900 words are entered which constitutes 288 groups of data, about:
Sample Number  IDNO1  Integer
Date           IDAT1  Integer
Beginning Time IBEGT1 Integer
Activity Code  IACTV1 Integer
Group Size Code IGSIZ1 Integer
Building Code  IBLDG1 Integer
Section Code   ISEC1  Alpha
Room Code      IROOM1 Integer
Suffix         ISSUF1 Alpha

According to the following format:

<table>
<thead>
<tr>
<th>3</th>
<th>1</th>
<th>4</th>
<th>2</th>
<th>1</th>
<th>3</th>
<th>2</th>
<th>4</th>
<th>1</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDNO1</td>
<td>IDAT1</td>
<td>IBEGT1</td>
<td>IACTV1</td>
<td>IGSIZ1</td>
<td>IBLDG1</td>
<td>ISEC1</td>
<td>IROOM1</td>
<td>ISSUF1</td>
<td></td>
</tr>
</tbody>
</table>

25 CHARACTERS

SECTION OF A RECORD/GROUP OF DATA

C)- Characteristics and formats of output. In order to use this tape for future programs, organization of the tape requires modification so that every group of data, defined above, can be handled as a complete record.

Characteristics of output tape are:
7 tracks
800 BPI
Record length = 4 words.
Block size = 255 records.
Label = DUKE2; binary

Output is composed by the following data:

<table>
<thead>
<tr>
<th>Sample Number</th>
<th>IDNO2</th>
<th>Integer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date</td>
<td>IDAT2</td>
<td>Integer</td>
</tr>
<tr>
<td>Sequence Number</td>
<td>ISEQ</td>
<td>Integer</td>
</tr>
<tr>
<td>Activity begining time</td>
<td>IBEGT2</td>
<td>Integer</td>
</tr>
<tr>
<td>Activity ending time</td>
<td>IENDT</td>
<td>Integer</td>
</tr>
<tr>
<td>Activity Code</td>
<td>IACTV2</td>
<td>Integer</td>
</tr>
<tr>
<td>Group Size</td>
<td>IGSIZ2</td>
<td>Integer</td>
</tr>
<tr>
<td>Building Code</td>
<td>IBLDG2</td>
<td>Integer</td>
</tr>
<tr>
<td>Section Code</td>
<td>ISEC2</td>
<td>Alpha</td>
</tr>
<tr>
<td>Room Code</td>
<td>IROOM2</td>
<td>Integer</td>
</tr>
<tr>
<td>Suffix</td>
<td>ISUFF2</td>
<td>Alpha</td>
</tr>
<tr>
<td>Space Type</td>
<td>ISPCT2</td>
<td>Integer</td>
</tr>
<tr>
<td>Room Use</td>
<td>IRMUS2</td>
<td>Integer</td>
</tr>
</tbody>
</table>

According to the following format:

---

7. Space type and room use are used as empty variables at this example, related information can be placed there later, in the available space.
I)- Input/Output designators. According to B-5500 systems, I/O designators are freely chosen, provided that these designators are kept consistent throughout the program. The following are the designators used in program:

**Input**: FILE 10, Name of file: DUKE1, Unit: TAPE

**Output**: FILE 6, Name of file: PRINT, Unit: LINE PRINTER

(This file is used for revision and diagnostical purposes).

FILE 12, Name of file: DUKE2, Unit: TAPE UNIT.

**Man hour distribution.**

A) Name of program : EHJ/DUKE2

Language: Fortran.

Estimated time : 35 minutes.

Compilation time : 13 secs.

8. It is normal however, though not restricted, that reader unit be assigned number 5 and line printer, number 6; this convention will be followed throughout the 2 cases.
Estimated core needs : 3712 words.

Description of function and solution method and special procedures are presented in CMHDIST/ARCH3.

C) - Assumptions made: Information about time is given in sixth of an hour (10 minutes); minutes range from 0 to 60, hours range from 0 to 24, days range from 1 to 7 and activities range from 1 to 24, one F.T.E = 168 man-hours per week.

D) - Limitations of the program. (See CMHDIST/ARCH3 discussion).

E) - User's options. The main option of this program relates to output, since for a quick insight into the actual situation, graphs can be omitted and only summary is obtained.

Another option can be introduced for testing purposes by entering a substantially reduced input, through punched cards.

F) - Characteristics and formats for input. Input has two forms:
   a- Magnetic tape.
   b- Punched cards.

Characteristics of magnetic tape #2, DUKE2 are:

7 tracks
800 BPI
Record length = 4 words.
Block size = 255 records.
Label = DUKE2; binary.

Data used from records refers to:

- Sample Number
- Day
- Begin hours
- Begin minutes
- End hours
- End minutes
- Activity Code

According to the following format:

Data entered from punched cards is used for headings and titles purposes, and is considered irrelevant for the process itself.
G) Characteristics and formats for output. All output is handled through line printer, length = 131 characters + 1 carriage control being displayed in two ways:

a) Bar graphs, and
b) Tables.

Bar graphs contain:

Hour of the day, man-hours at that hour, and the graph itself, the reason for this format is a quick reference between actual man-hour figure and the bar that represents it.

Fields for output are alphanumeric and floating point values.  

Table contains:

Activity, man-hours total, percent with respect to activity, percent with respect to day, for each day, and a final column with total man-hours per activity.  

1) Input/Output designators.

Input: FILE 5, Name of file: DUKE2, UNit: CARD READER

FILE 3, Name of file: DUKE1, Unit: MAGNETIC TAPE

9. See format specifications figure in CMHDIST/ARCH3.

10. See format specifications figure in CMHDIST/ARCH3.
J)- Error messages. Whenever a bad information is found in a record the record is printed, with the message: "BAD RECORD" attached to it. Information is not processed and new record is read.

Preliminary statistical analysis.
Frequency distribution, central tendency measures.

A)- Name of Program : EHJ/DUKE3.
Language : Fortran
Estimated time : 20 minutes.
Compilation time : 27 secs.
Estimated core storage : 5568 words.

Program used to read data from tape #2, about beginning and ending times and activities, processing the data and producing information about frequency distribution, according to a given interval (10 minutes in this example) and display bar graphs showing the most common intervals used.

Computations are carried out in order to obtain values of central tendency measures, i.e.: mean, median and mode.
Further developments based upon these program results will provide means for determining degrees of confidence, for decisions about scheduling and space allocations.

B)- Solutions method and procedure. Data to be processed is
considered in two different forms:

a) - Non-grouped.
b) - Grouped.

a) - Non-grouped data stands for the case where each datum is considered independently in the set of numbers called "sample space"

b) - Grouped data, stands for the case where items are considered together within a subset called interval class, of the set of numbers in the sample space. Interval class size for this example is 10 minutes, and since data was recorded rounded to the nearest 10 minutes, it is considered non-grouped.\(^{11}\)

One of the central features of this program is this interval class size, which conceptually applies to both cases, and based upon it, the entire process is performed. The other important feature is an array PQ whose parameters are given by activities and number of intervals, used to store values of frequencies.

A group of variables are set for:

<table>
<thead>
<tr>
<th>Description</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of activities</td>
<td>NACT</td>
</tr>
<tr>
<td>Number of intervals</td>
<td>NINT</td>
</tr>
<tr>
<td>Number of intervals per hour</td>
<td>NIN</td>
</tr>
</tbody>
</table>

\(^{11}\) Same data, when interval class size is increased, necessarily changes to grouped.
Size of interval : SINT
                : ISINT

As well as initial boundaries for first interval:

Lower bound   : XINT (1,1)
Upper bound   : XINT (1,2)

Logical control of the process (enabling the program to
take care of grouped and non-grouped data) is controlled by
CASE1. Other variable names used are:

Mode          : XMOD
Cumulative frequency : CFQ
Percent of frequencies with
respect to total CFQ : PACT
Middle point of interval
class          : MPT
Lower bound   : LB
                : MEAN
                : MEDIAN

At the outset arrays and variables are cleared to avoid the
use of other values which might remain in such locations.
Lower, middle and upper points of each one of the intervals
are computed, and Data from tape #2, ARCH3 about:

Activity Code, beginning and ending times are entered.
Check is made for beginning and ending times to test validity of information and number of intervals relating to hours and minutes are computed and sorted temporary as:

TEMP1, for hours.
TEMP2, for minutes.

which properly combined, determine the ordinal number of the interval associated to each activity, which is placed in another temporary location INT.

Range of values for INT is 1 to 48, which is checked; two actions are taken:

a) INT. EQ. 0, restart reading cycle.
b) INT. GE.48, INT = 0

Value of INT associated with the activity performed, becomes an actual parameter of a particular element of FQ array, and its occurrence increases by one the number of occurrences of that particular situation.

Here, the use of FQ becomes clear: It is used as an "ocurrences counter", which in fact is the meaning of "frequency". The process is repeated until input tape is exhausted, at which point each element of the FQ array contains the final number of occurrences of each activity, according to each particular interval.

The process continues by computing cumulative frequencies
(CFQ), mean, median and mode (XMOD) branching to actions that consider each one of the two mentioned cases: non-grouped and grouped data.

Solution methods for CFQ, mean, median and mode.

for \( j=1,2,\ldots,11 \)

\[
\text{CFQ} = \sum_{i=1}^{i=n} FQ_{j,i} \quad (1)
\]

\[
\text{MEAN} = \sum_{i=1}^{i=n} \frac{FQ_{j,i} \cdot X_i}{N} \quad (2)
\]

\( j \) refers to activities.

Where \( X_i = \) Middle point (MPT) for grouped data.

Sample point (MPT) for non-grouped data. 12

\[
N = \text{Total cumulative frequency}
\]

For \( j=1,2,\ldots,11 \)

\[
\text{MEDIAN} = \text{LB} + \text{SINT} \cdot \frac{\text{TFQ/2}}{\sum_{i=1}^{i=n} \frac{\text{CFQ}_b}{FQ_m}} \quad (3)
\]

Where

\( \text{LB} \) : Lower class boundary of the class containing the median.

\( \text{CFQ}_b \) : Cumulative frequency below the interval class containing the median.

\( FQ_m \) : Frequency of the interval class containing the median.

12. MPT is a variable name bearing both concepts in order to simplify some program.
SINT : Interval size of the class containing the median.
MODE : Items with the higher frequency (for non-grouped data).

MODE : \( \text{LB+SINT}^* \frac{A1}{A1+A2} \) (for grouped data) \( (4^1) \)

Where:

LB : Lower class boundary of the class containing the mode.
A1 : Maximum frequency - Frequency above the interval class containing the mode (\( F\text{QMX} - FQ_{i+1} \))
A2 : Maximum frequency - Frequency below the interval class containing the mode (\( F\text{QMX} - FQ_{i-1} \)).\( ^{13} \)

Procedure: After reading cycle is completed, cumulative frequency and mode are computed in a common iteration; determining next, a normalized (as percent of frequency about total) value of frequency which is used as parameter of bars in the graph (process for producing the bar graph is discussed in \text{CMHDIST}/ARCH3, for reference, note however that while that program produces all graphs at a time, the present produces one graph, and a summary at a time).

After each graph is produced, the program starts a search for the interval class containing the median; this step is carried out with a temporary variable DUM2, which has the

---

13. Detailed discussion about those methods may be found in Hays and Spiegel.
value of half total cumulative frequency and interval class containing median. At this point, method #3 is applied for median final computation.

Finally, computations for mode start by setting a group of auxiliary variables NN, N, IK, and IM, used in determining whether mode exists or is unique; another variable FQMX is used to store the highest value of all frequencies, determined by comparisons and simultaneously find the interval class containing the modal value.

Control of the program is taken by CASE1, for grouped or non-grouped data.

For non-grouped data, the criterion stated in method (4) is followed, otherwise, method (4\textsuperscript{1}) is applied, and mode is computed.

Computations for mode are repeated when it has been found that it is not unique.

Results about mean, median and mode(s) are printed below each graph.

C) Assumptions made:

Number of activities (NACT) : 24
Number of intervals (NINT) : 48
Number of intervals/hour (NIN) : 6
Size of interval (SINT, ISINT) : 10
Lower and upper limits for first interval

\[
\begin{align*}
XINT (j,1) & : 1 \\
XINT (j,2) & : 10
\end{align*}
\]

Data non-grouped (i.e. CASE1=TRUE)

D)- Limitations of the program. Input: Form is fixed by DUKE2 TAPE.

E)- User's options:

Initial values for:

- Number of activities (NACT)
- Number of intervals (NINT)
- Size of interval (SINT)

may be modified according to actual problem.

Note that any modification of size of interval has a direct influence in the way data is considered, and control switch CASE1 as well has to be modified by the user. Options in output are restricted by the fact that graph representation provides a very clear representation about the behavior of the frequency distribution curve.

F)- Characteristics and formats of input.

Magnetic tape #2 is used as input, with the following characteristics:

7 tracks
800 BPI
Record length = 4 words.
Block size = 255 records.
Label = DUKE2, binary.

Data entered from records, refer to:

Begin hours  HOUR  Integer
Begin minutes  MIN  Integer
End hours  NDHR  Integer
End minutes  NDMIN  Integer
Activity code  IACTV  Integer

According to the following format:

![Diagram]

DATA ENTERED FOR THIS PROGRAM

Data entered from punched cards is used for headings and titles purposes, and is considered irrelevant for the process itself.
G)- Characteristics and formats for output. All output is made by means of the line printer, with 131 characters length plus 1 carriage control.

Information is displayed in two forms:

a) - Bar graphs,
b) - Tables of measures values.

Bar graphs contain:

Interval, frequency, cumulative frequency, percent, and bar graph itself; also this form is chosen to provide a quick reference between actual values of frequencies and distribution's curve shape.

<table>
<thead>
<tr>
<th>Actual interval</th>
<th>XINT</th>
<th>Real</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency</td>
<td>FQ</td>
<td>Real</td>
</tr>
<tr>
<td>Cumulative frequency</td>
<td>CFQ</td>
<td>Real</td>
</tr>
<tr>
<td>Percent of frequency relative to total frequency</td>
<td>PACT</td>
<td>Real</td>
</tr>
<tr>
<td>Graphical value of percent</td>
<td>XLINE</td>
<td>Alpha</td>
</tr>
</tbody>
</table>

According to the following format:
Tables contain:

Mean, median and mode, printed names, with actual values.
These tables are printed next to the corresponding graph also for immediate reference.

Mean          MEAN          Real
Median        MEDIAN        Real
Mode          XMOD          Real

According to the following format:
### Activity

**Mean** | **Median** | **Var** | **Stdv** | **Xmod(s)**
---|---|---|---|---
16 | 12 | 10 | 12 | 10 | 12 | 10 | 12

### Sample of Table Format

1) Input/Output designators.

**Input**: FILE 5, Name of file: DUKE4, Unit: CARD READER

**Output**: FILE 6, Name of file: PRINT, Unit: LINE PRINTER

2) Error messages. Whenever bad data of any kind are found in input records, this record is printed with the message "Bad record" attached. In such a case, information is not processed, and a new record is read.

Complete listings, flowcharts and samples of output are provided in part V, appendices, for programs labeled EHJ/DUKE2, and EHJ/DUKE3, consult flowcharts under CMHDIST/ARCH3 and DMHDIST/ARCH4 labels respectively. Those flowcharts essentially deal with the logic followed so far.
The following set of programs, developed based upon the experience obtained in former example, deal with the same kind of problems, and some extra alternatives are studied. More systematic and accurate ways for selecting the information are developed, and the scope of the case of preliminary statistical analysis is broadened.

Nature of data collected at Rice.

As a means of applying the experience just studied, and combined with a course on computers at the school of Architecture of Rice University during the fall of 1969, it was decided to perform an experiment of the process just studied with students attending the course, from the second and fifth years.

Some considerations about the sample must be kept in mind before going through the process:

A) Samples were selected from a very defined subset of the Rice population, and not at random; consequently the information to be obtained does not reflect the campus population as a whole, whereas the spaces considered, represent to a greater extent, the whole campus.

B) The analysis of space was carried out up to the point of designating the particular place; no consideration about room, sections or any other subcategories was made.
A set of twenty-six spaces, which were the most common found from students' information, is next coded; it does not represent the entire campus, but the spaces considered add up to more than 80% of the whole.

In order to place the code, columns 71-72 were selected in order to make use of the cards, punched by the students and process code of spaces, together with the rest of the information.

C)- Preliminary data preparation was carried out by the students. By means of a simplified diary and as part of their training in the course the card punching was carried out by them. Some unavoidable errors were expected, which the author tried to minimize through the next steps, and reduced to less than 3 percent.

The experience obtained is considered, however, fruitful, and strong enough to substantiate statements made in this work and demonstrates that further developments, deeper and broader, are the most powerful tool in the decision-making process of campus planning.

As already mentioned, data collected were recorded by the students in a diary form, which they handled during a week. The form has the following format:
Students filled the entries in the table according to these specifications. Location was not coded at this step, but later, in the next steps. (This was an option taken. Obviously any information recorded on diaries can be coded the first time.) The reason for this was that it was intended to make the students' exercise as clear as possible.

This information is transferred to punched cards, with the same format:

<table>
<thead>
<tr>
<th>Sample number</th>
<th>Columns 1 - 3</th>
<th>Integer value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activity</td>
<td>Columns 5 - 6</td>
<td>Integer value</td>
</tr>
</tbody>
</table>

DIARY SAMPLE
Day          Columns  8 - 9  Integer value
Begin time   Columns 11 -14  Integer value
Number in group  Columns 16 -18  Integer value
Number in space   Columns 20 -23  Integer value
Location      Columns 24 -70  Alphanumeric.

As shown:

Data preparation, step 1. Prepare magnetic tape from punched deck.

A)- Name of the program : AMHDIST/ARCH1
Language        : Fortran
Estimated time  : 6 minutes.
Compilation time : 8 secs.
Estimated core storage : 4224 words.

Program used to read data cards produced by the students.

The data deck was sorted first, in an IBM sorter, in order to have it ordered by sample numbers, days, hours and numbers. No actual processing is made with the data, but a set of checks is made prior to transferring each record (card) to the output device, which is a magnetic tape, to be created.

B)- Solution method and special procedures. The essential feature of this program are a set of tests, made in order to "clean" data of errors. A set of variables representing sample size, number of activities and number of spaces is given at the outset.

Each card is read for: Sample number, activity code, date, beginning time, group size, number of people in space, and space code.

Check is made for: Sample number, activity code, date, begin time and space code.

Any information outside the allowable range causes the program to print the record and read the next one.

Sample number 4, was randomly selected, in order to print it as a means of reference.
C) Assumptions made are:

Sample size : 1 - 43 samples
Number of activities : 1 - 11
Number of spaces : 1 - 26

They represent the allowed range for the corresponding variables, also it is assumed that ranges of
Day : 1 - 7
Time : 0000 - 2400

Data was collected in general by five minutes (this assumption will be referred to when it affects decisions on further programs)

D) Limitations: The program is used only to transfer data from cards to tape and perform the first of a series of tests designed to clean the data; those tests will be made all the way through the processing programs. The aim is to conclude with the most reliable information as possible.

E) User's options. The program is designed in such a way that by making minimum changes it is able to handle any desired values, for sample size, number of activities and number of spaces.

It reads the information in the order above mentioned but the user has the option to compact it as much as he wishes to, according to formats provided for each case.

Output options are restricted by the most efficient use of
magnetic tape, as B-5500 systems handle them.

F)- Characteristics and formats for input. Input source is a deck of punched cards to be read.

Data entered from records refer to:

<table>
<thead>
<tr>
<th>Sample number</th>
<th>IDNO</th>
<th>Integer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activity code</td>
<td>IACTV</td>
<td>Integer</td>
</tr>
<tr>
<td>Date</td>
<td>IDAT</td>
<td>Integer</td>
</tr>
<tr>
<td>Begin time</td>
<td>IBEGT</td>
<td>Integer</td>
</tr>
<tr>
<td>Group size</td>
<td>IGSIZ</td>
<td>Integer</td>
</tr>
<tr>
<td>People in space</td>
<td>ISNUM</td>
<td>Integer</td>
</tr>
<tr>
<td>Space code</td>
<td>ISPAC</td>
<td>Integer</td>
</tr>
</tbody>
</table>

According to format described in data card sample.

G)- Characteristics and formats for output. Output is produced in two forms:

a)- Through line printer for error messages, and reference sample, and

b)- Through magnetic tape unit, for tape generation, containing information only about records found correct.

Optimum use of magnetic tape is intended; and its characteristics are:

7 tracks
800 BPI
Record length = 3 words (8 characters per word)
Block size = 341 records \((341 \times 3 = 1023)\), which is exactly the maximum length permitted by B-5500 system.

Label = ARCH2, binary.

Data are placed in the record on tape about: same list of data described in (F), according to the following format:

```
+--------+--------+--------+--------+--------+--------+--------+
| IDNO   | IACTV  | IDAT   | IBEG   | ISIG   | ISNUM  | ISPAC  |
+--------+--------+--------+--------+--------+--------+--------+
| 3      | 2      | 2      | 4      | 3      | 3      | 7      |
+--------+--------+--------+--------+--------+--------+--------+
```

3 WORDS \((8\text{ CH. EA.) : 24\text{ CHARACTERS}}\)

ORIGINAL RECORD/GROUP OF DATA

\begin{itemize}
  \item[I)] Input/Output designators, are:
  \begin{itemize}
    \item Input : FILE 5, Name of file : ARCH1, Unit: CARD READER
    \item Output : FILE 6, Name of file : PRINT, Unit: CINE PRINTER.
      (used for reference sample output, and error messages)
  \end{itemize}
  \item[FILE 10, Name of file : ARCH2, Unit: MAGNETIC TAPE
  \end{itemize}

\item[J)] Error messages. Whenever wrong data is read from records on punched cards, concerning sample size, number of activities
number of spaces, days or time, an error counter is increased by one, the record is printed along with the message of "bad record" attached to it, and no process is performed with those items; a new record is read, and tests are repeated.

Complete listings and output samples are found in appendices.

Data preparation, step 2.

A) Name of the program : BMHDIST/ARCH2
Language : Fortran
Estimated time : 6.25 min.
Compilation time : 13 secs.
Estimated core : 6656 words.

This program is used to read from tape #1 ARCH2, all information contained in records, perform a second series of tests of data, and by means of a series of comparisons between records determine the ending time of each activity performed by each sample, as well as sequential activity numbers by days.

Output of this program is a new tape #2, ARCH3, with new organization of records, owing to data added about ending time and sequential numbers. New records are organized in order to be used as input of next programs.
B) Solution method and procedures. Data are read from input tape #1, and each record just read is placed in temporary locations, and a new record is read. The process has the purpose of always comparing two sequential records.

Check about equality of sample numbers, dates, and beginning time is performed, in order to determine ending times for each activities. The process is carried out by assigning as end time of an activity, the beginning time of the next activity; or by assigning an end time of 2400 when sample numbers or days change, in which case, a new record is generated in order to have information for each day and change of sample completed.

When there is a change of sample, a new record is read, since the next record belongs to a new sample.

When there is a change of days, i.e., an activity spanning from one day to the next, this activity is considered ending at 2400 for first day, and ending at the beginning time of the next activity of the second day.

In each of the above cases, when the beginning time is greater than or equal to the ending time the record is wrong, an error counter is increased by one, and the record is printed for reference, but not transferred to the tape, and the process continues until the input tape is exhausted.

C) Assumptions. A particular sample is printed on the line
printer, for the user's reference, in the example. Sample numbers greater than 36 were selected for this purpose.

D) Limitations. Input is restricted in form by tape generated in AMHDIST/ARCH1. The program is limited to a particular kind of process, but is also part of a set of programs for analyzing information as a whole process.

E) User's options. There is no limit or restriction of the program to the amount of data handled. Options for output are restricted to the optimum use of output tape #2.

F) Characteristics and formats of input.

Magnetic tape #1, ARCH2, is used as input for this program, and its characteristics are:
7 tracks
800 BPI
Record length = 3
Block size = 341
Label = ARCH2, binary.

Data entered from records, refers to the same list of data described in AMHDIST/ARCH1 output. According to the following format:

```
<table>
<thead>
<tr>
<th>IDNO</th>
<th>IACTV</th>
<th>IDAT</th>
<th>IEEEI</th>
<th>IEGSI</th>
<th>IGSIS</th>
<th>ISNUM</th>
<th>ISPAC</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td></td>
<td>7</td>
</tr>
</tbody>
</table>
```

*3 WORDS *(8 CH. EA.)* : 24 CHARACTERS

ORIGINAL RECORD/GROUP OF DATA
G)- Characteristics and format of output. Output is produced in two forms:

a)- Through line printer for error messages and reference sample.

b)- Through magnetic tape unit for tape generation, containing information only about records found correct.

Optimum use of tape is intended, and its characteristics are:

- 7 tracks
- 800 BPI
- Record length = 4 words.
- Block size = 255 records.
- Label = ARCH3, binary.

Data are placed in the record, on tape about:

<table>
<thead>
<tr>
<th>Sample number</th>
<th>IDNO2</th>
<th>Integer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activity code</td>
<td>IACTV2</td>
<td>Integer</td>
</tr>
<tr>
<td>Date</td>
<td>IDAT2</td>
<td>Integer</td>
</tr>
<tr>
<td>Sequential number</td>
<td>ISEQ</td>
<td>Integer</td>
</tr>
<tr>
<td>Begin time</td>
<td>IBEGT2</td>
<td>Integer</td>
</tr>
<tr>
<td>End time</td>
<td>IENDT</td>
<td>Integer</td>
</tr>
<tr>
<td>Group size</td>
<td>IGSIZ2</td>
<td>Integer</td>
</tr>
<tr>
<td>People in space</td>
<td>ISNUM2</td>
<td>Integer</td>
</tr>
<tr>
<td>Space code</td>
<td>ISPAC2</td>
<td>Integer</td>
</tr>
</tbody>
</table>

According to the following format:
I)- Input/Output designators are:
Input : FILE 10, Name of file : ARCH2, Unit : MAGNETIC TAPE
Output : FILE 12, Name of file : ARCH3, Unit : MAGNETIC TAPE
FILE 6, Name of file : PRINTER.Unit : LINE PRINTER
(used for error messages and reference sample output).

J)- Error messages. Whenever beginning time is greater than or equal to ending time, an error counter is increased by one, the record is printed along with a message "bad record" attached to it and no transference of the record is made to output tape.

Complete listing and samples of output are found in appendices.

Man-Hour distribution.

A)- Name of program : CMHDIST/ARCH3
Language : Fortran
Estimated time : 6.20 minutes
Compilation time : 45 secs.
Estimated core requirements : 5568 words.

Program used to read data from tape #2, about samples, activities, days and hours; processing these data and producing information about the activities in which students are involved (the measure units are man-hours) the time devoted to each activity in units of man-hours, the particular hour, and day related to.

Information is produced by the computer by means of bar graphs and a table containing summarized information.

The purpose of the bar graph representation is to show the hours of the day, at which particular activities have their peaks, and displayed in such a way that a quick look can provide a close idea of the actual situation on campus.

B)- Solution method and special procedures.

A three-dimensional array count is set with parameters given by: activities, days and hours. This counter is the central element of the program. Any quantity stored there, will be related to a particular activity, day and hour.

Other arrays are designated in which totals will be stored, related to:
Activity and day = TOT
Activity = ACTOT
Day = DYTOT
Big total = BIGTOT

All these elements will be used in final summaries.

Data about: Sample number, day, begin hours and minutes, end hours and minutes and activities are read in, from input tape #2.

Check is made for:
Activity codes
Day numbers
Beginning times, and
Ending times.

The first of those items found outside allowed range, an error counter is increased by one, the bad record is printed and a new one is read again.

If all the items are correct, processing of the data starts.

The range of values of variables read is from one to some fixed number according to the case studied, except for begin and end hour, which range is from 0 to 24, and since activities, days and hour will be used as parameters of arrays, the last has to be modified since Fortran language limits the range of any parameter of an array to be equal to, or greater than one. This limitation is easily handled, by increasing
by one unit, every begin and end hours. New values of begin hour and end hour are compared and according the relative result of this comparison (less than, equal to or greater than) the program branches into one of three different procedures in order to obtain the amount of man hours devoted to a particular activity, a particular day at any particular time (since every record read contains information about those aspects). This counting process is carried out in hours and fractions thereof, and each value thus computed is stored in its corresponding element of COUNT array.

The same information is taken for totals and the corresponding elements are increased by the amount contained in the central counter.

This process is repeated until the input tape is exhausted. At this point, arrays will have final values of the corresponding items.

A graph is prepared by means of an array called LINE in order to store there:

a) Blank spaces and reference dots.

b) Values of man-hours, rounded off to the nearest integer value, represented by character X.

Line is printed together with values of hour, and values stored in corresponding COUNT's array elements.

Process is repeated until a complete bar graph has been plotted
placing two graphs per page.

Each graph represents a particular activity for each day of the week. At the end of the process percentages are computed referring to total man-hours with respect to total man-hours per activity, and total man-hours per day. Those results are printed in the summary table.

C) - Assumptions made. Information about time is given in twelfths of an hour (5 minutes), minutes range from 0 to 60, hours range from 0 to 24, days range from 1 to 7, and activities range from 1 to 11. One FTE is equal to 168 man-hours per week.

D) - Limitations. Input is limited to the format provided by input tape #2, ARCH3. Program processes information in a particular way, and therefore is limited to such particular case, however this limitation is counterbalanced by the fact that the program belongs to a group of programs intended to provide a comprehensive analysis of information about the particular case studied, and which output is just a part of a complete set of results.

E) - User's options: (See EHJ/DUKE2).

F) - Characteristics and formats for input. Magnetic tape #2, ARCH3 is used as input, its characteristics are:

7 tracks
800 BPI

Record length = 4
Block size = 255 records.
Label = ARCH3; binary.

Data entered from records refers to:
Sample number ID Integer
Activity code IACT Integer
Day DAYCD Integer
Begin hours HOUR Integer
Begin minutes MIN Integer
End hours NDHR Integer
End minutes NDMIN Integer

According to the following format:

```
<table>
<thead>
<tr>
<th>ID</th>
<th>IACT</th>
<th>DAYCD</th>
<th>HOUR</th>
<th>MIN</th>
<th>NDHR</th>
<th>NDMIN</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>
```

4 WORDS

DATA ENTERED FOR THIS PROGRAM

Data entered from punched cards is used for headings and titles purposes, and is considered irrelevant for the process itself.
G) Characteristics and formats for output. Output is completely carried out through the line printer, in two different forms:

a) Bar graphs

b) Summary table.

Bar graphs contain:

Hour of the day, man-hours at that particular hour, and the graph itself, the reason for this format is a quick reference between actual man-hour figure and the bar representing it.

<table>
<thead>
<tr>
<th>Hour of the day</th>
<th>XTIME</th>
<th>Real</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of man-hours</td>
<td>COUNT (array)</td>
<td>Real</td>
</tr>
<tr>
<td>Graphic representation of man-hours</td>
<td>XLINE</td>
<td>Alpha</td>
</tr>
</tbody>
</table>

According to the following format:

```
<table>
<thead>
<tr>
<th>ACTIVITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>XTIME</td>
</tr>
<tr>
<td>COUNT</td>
</tr>
<tr>
<td>XLINE</td>
</tr>
</tbody>
</table>
```

![Sample bar graph format]
Table contains:

Activity, total man-hours, percent with respect to activity, percent with respect to day, for each day; and a final column with total man-hours per activity.

<table>
<thead>
<tr>
<th>Activity name</th>
<th>ACTNAM</th>
<th>Alpha</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total man-hours per activity/day</td>
<td>ITOT</td>
<td>Real</td>
</tr>
<tr>
<td>Percent of man-hours relative to total per activity</td>
<td>PCTAC</td>
<td>Real</td>
</tr>
<tr>
<td>Percent of man-hours relative to total per day</td>
<td>PCTDY</td>
<td>Real</td>
</tr>
<tr>
<td>Total per activity</td>
<td>ACTOT</td>
<td>Real</td>
</tr>
<tr>
<td>Total per day</td>
<td>DYTOT</td>
<td>Real</td>
</tr>
</tbody>
</table>

According to the following format:

<table>
<thead>
<tr>
<th>ACTIVITY</th>
<th>MONDAY</th>
<th>FRIDAY</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTNAM</td>
<td>ITOT(1)</td>
<td>PCATC(1)</td>
<td>PCTDY(1)</td>
</tr>
<tr>
<td>1</td>
<td>12</td>
<td>2</td>
<td>5</td>
</tr>
</tbody>
</table>

SAMPLE OF TABLE FORMAT
I)- Input/Output designators:

Input : FILE 5, File name : ARCH4, Unit : CARD READER
       FILE 3, File name : ARCH3, Unit : TAPE
Output : FILE 6, File name : PRINT, Unit : LINE PRINTER

Preliminary statistical analysis.

A)- Name of program : DMHDIST/ARCH4
Language : Fortran
Estimated process time : 3 minutes
Compilation time : 14 secs.
Estimated core storage : 2496 words.

Program used to read data from tape #2, ARCH3 concerning activities, beginning and ending times; processing the data, and producing information about frequency distribution, according to a given interval (for this case 10 minutes) and displaying bar graphs.

Computations are carried out in order to obtain values of central and dispersion measures, i.e., mean, median, mode, standard deviation and variance. Further developments based upon the results obtained in this program will provide means for determining the degree of confidence for decisions about scheduling and space allocation.

B)- Solution method and special procedures. Interval size of 10 minutes is one of the central elements of the program.
Based upon that, all processes are performed. An integer array called FQ, which parameters are activities and numbers of intervals, constitutes the second important element of the program.

A group of variables are set for:
- NACT : Number of activities.
- NINT : Number of intervals.
- NIN  : Number of intervals per hour.
- SINT : Size of intervals.

and initial boundaries for first interval XINT (1,1), XINT (1,2) logical control for the process is controled by switch CASE1.

Other variables used are:
- XMOD : Mode
- CFQ  : Cumulative frequency
- PACT : Percentage of frequencies to total CFQ
- MPT  : Middle point
- LB   : Lower bound
- VAR  : Variance
- STDV : Standard deviation

At the outset subscripted variables are cleared and all values of lower, middle and upper are computed for the intervals and data stored on tape #2 ARCH3 are read in.

IACT : Activity code
HOUR, MIN. : Beginning time
NDHR,NDMIN : Ending time

Check is made for ending and beginning time. If end time is
less than or equal to begin time, record is printed for
report and a new one is read, otherwise the process con-
tinues by placing in temporary locations (TEMP1, TEMP2) the
number of intervals for hours and minutes; this will de-
termine the interval ordinal number associated to each ac-
tivity. For the computations of intervals in minutes, the
program considers if the time involved is between 0 and 10
minutes and if so adds 0.99 to the results (so that when
this value is converted to an integer it will have a value
of 1).

Total number of intervals (INT) is the sum of the number of
intervals for hours and minutes, and is checked for two
conditions:
a)- INT = 0, no time is involved, error condition is consider-
ed, process is repeated.
b)- INT.GT.NINT, time involved exceeds 8 hours, INT is fixed
to that limit.

Once information about a number of intervals is ready and the
correctness of computations is checked, those values are
used as parameters of elements of the array FQ and this fact
accounts for an occurrence of an interval. This process is
repeated until the input tape is exhausted, at the end of
which all elements of FQ contain the total number of occurrences of intervals-activities; those FQ elements are considered as the sample points for the next statistical computation.

Process continues by computing the cumulative frequency (CFQ), from frequencies, switch CASE1 branches to one of two different parts of the program for conditions:

a) Non-grouped data, CASE1 : TRUE
b) Grouped data, CASE1 : FALSE

according to the condition encountered, mean, median and mode, variances and standard deviation, are computed according to:

\[
\text{CFQ} = \sum FQ
\]
\[
\text{MEAN} = \frac{\sum FQ \times X}{N}
\]

Where: 
- \( X \): Middle point (MPT) for grouped data
- \( X \): Sample point (MPT) for non-grouped data
- \( N \): Total cumulative frequency (TFQ)

\[
\text{MEDIAN} = \text{LB} + \text{SINT} \times \left[ \frac{N/2 - \sum fb}{\sum fm} \right]
\]

* for grouped data.

\[
\text{MEDIAN} = \begin{cases} 
\text{half way between two central values.} & \text{if number of sample points is even} \\
\text{simply the central value.} & \text{if number of sample points is odd}
\end{cases}
\]

* for non-grouped data.

Where \( \text{LB} \): Lower interval bound of the class containing the
median.

\[ fb = \text{Cumulative frequency below the interval class containing the median (CFQ)} \]

\[ fm = \text{Frequency of the class containing the median (FQ)} \]

\[ \text{SINT} = \text{Interval class size} \]

\[ \text{MODE} = \text{LB} + \text{SINT} \times \frac{A1}{A1 + A2} \] (4)*

* for grouped data.

\[ \text{MODE} = \text{Value of the most frequent point} \] (4*1)*

* for non-grouped data.

Where: \( A1 = \text{Maximum frequency (FQMX)} - \text{frequency above the class containing the mode (FQ)} \)

\[ A2 = \text{Maximum frequency (FQMX)} - \text{Frequency below the class containing the mode (FQ)} \]

\[ \text{VARIANCE} = \frac{\sum FQ \times X^2}{N} - \bar{X}^2 \] (5)

Where: \( X = \text{Middle point (MPT) for grouped data} \)

\( \bar{X} = \text{Sample point (MPT) for non-grouped data.} \)

\( \bar{X} = \text{Mean} \)

\[ \text{STD.DEV.} = \text{Square root of variance} \] (6)

Procedure: Mean, variance and std.dev., are computed in the same iteration, after which frequencies are normalized (as percentages of the total frequency). This normalized value is represented as bar graph of frequency distribution.

In next step search for the interval class containing the median starts and switch CASE1 controls the program for
grouped or non-grouped data. When data is grouped, median is computed according to (3), otherwise a check is made to determine whether or not the number of points in the sample is even (ITEST) and determining the median according to (31). Computation of mode starts by setting a group of variables, NN, N, IK and IM used to check and compute multi-modal cases, determine the highest value of frequencies and the interval class containing the modal value. (FQMX)

Switch CASE1 controls branching for grouped or non-grouped data! for grouped data checks whether or not mode exists and whether it is contained in first or last intervals, and finally mode is computed according to (4). For non-grouped data existence of mode is checked and mode is computed according to (41). For both cases, a check is made to determine uniqueness of mode.

When the process is finished all results are printed after each table.

C)- Assumptions made:

Number of activities : 11
Number of intervals : 48
Size of interval : 10 minutes
Number of intervals/hour : 6
Data is considered grouped and recorded by 5 minutes.

D)- Limitations. Input forms are limited to format provided by tape #2 ARCH3.
E) - User's options. All values assumed in C can be modified according to user's needs. Data is considered grouped or non-grouped according to collections criterion and interval size selected.

F) - Characteristics and formats of input. Magnetic tape #2 ARCH3 is used for input.

7 tracks
800 BPI

Record length = 4 words.
Block size = 255 records.
Label = ARCH3, binary.

Data entered from records refers to:

Activity code IACT Integer
Begin hours HOUR Integer
Begin minutes MIN Integer
End hours HOUR Integer
End minutes NDMIN Integer

According to the following format:

DATA ENTERED FOR THIS PROGRAM
Data entered from punched cards is used for headings and titles purposes, and is considered irrelevant for the process itself.

G) - Characteristics and formats for output. Output is considered in two forms:
   a) - Bar graphs, and
   b) - Tables of values of measurements

Bar graphs contain:
Interval, frequency, cumulative frequency, percent, and the bar graph itself; also this form is chosen to provide a quick reference between actual values of frequencies and its distribution curve shape.

<table>
<thead>
<tr>
<th>Actual interval</th>
<th>XINT</th>
<th>Real</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency</td>
<td>FQ</td>
<td>Real</td>
</tr>
<tr>
<td>Cumulative frequency</td>
<td>CFQ</td>
<td>Real</td>
</tr>
<tr>
<td>Percent of frequency relative to total frequency</td>
<td>PACT</td>
<td>Real</td>
</tr>
<tr>
<td>Graphical value of percent</td>
<td>XLINE</td>
<td>Alpha</td>
</tr>
</tbody>
</table>

According to the following format:
Tables contain:
Mean, median, variance, standard deviation, and mode actual values, these tables are printed next to the corresponding graph for immediate reference.

Mean \hspace{1cm} \text{MEAN} \hspace{1cm} \text{Real}
Median \hspace{1cm} \text{MEDIAN} \hspace{1cm} \text{Real}
Variance \hspace{1cm} \text{VAR} \hspace{1cm} \text{Real}
Standard deviation \hspace{1cm} \text{STDEV} \hspace{1cm} \text{Real}
Mode \hspace{1cm} \text{XMOD} \hspace{1cm} \text{Real}

According to the following format:
I) - Input/Output designators.

Input : FILE 5, File name : ARCH5, Unit : CARD READER
       FILE 3, File name : ARCH3, Unit : TAPE
Output : FILE 6, File name : PRINT, Unit : LINE PRINTER

J) - Error messages. Whenever wrong information is found in a record, it is printed for reference. No use is made of it.

MHDIST (II)

A) - Name of program : EMHDIST/ARCH5
Language : Fortran
Estimated process time : 6.5 minutes
Compilation time : 11 secs.
Estimated core : 5696 words.

The program is used to read data from tape #2 ARCH3, about
sample numbers, days, begin and end times, and spaces, processing and producing information about amounts of man-hours per space each day of the week, regardless of the kind of activity performed there. Shows "loads" in terms of man-hours in spaces in order to study each particular situation as space demand and peaks in the time of this demand.

A summary table is printed which shows the amounts of man-hours per space for all days of the week, and the percentage relative to total man-hours per day and per activity.

B)- Solution method and special procedures. A three-dimensional array CNT with parameters space, day and hour, is the central element of the program, stored the values of man-hours related to each particular space, day and hour and is used to produce the bar graphs and summary table.

Procedure. Read in data from tape #2, ARCH3 concerning: students' information, check validity of data testing minutes within allowed range (0-60 minutes), activities and spaces were already checked. Invalid data is printed for reference. Valid data enters the process, which first step consists of increasing begin and end hours by one, to be used as parameters of CNT.

Ending and beginning hours are compared and according to the logical result program branches into one of three different blocks where man-hours are computed and stored at CNT. Each
block takes care of one of those cases:

1) \text{End hour .LT. begin hour}
2) \text{End hour .EQ. begin hour}
3) \text{End hour .GT. begin hour}

computing man-hour figures accordingly.

Values of man-hours resulting are stored in array count and
program computes summaries according to space and day, space,
and day, and finally big total.

Totals are stored in secondary arrays according to the
order mentioned.

The process recycles until input tape has been exhausted and
arrays contain final figures. At this point, a cycle is
started to produce the graphs. After graphs are printed, the
program produces the summary table determining values for
man-hours per day for each space (TOTSDO, and computing their
values relative to total man-hours per day (TOTDY), and total
man-hours per space (TOTSP) expressed as percentages
(PCTDY and PCTSP), respectively.

C) Assumptions made. Information about time is given by 5
minutes, range of values for begin and end hours is 0 to
2400, and range of values for begin and end minutes is 0 to
60, days range from 1 to 7, and spaces range from 1 to 26.

D) Limitations. Input is limited to the format provided by
tape \#2, ARCH3.
E) - User's options. Number of activities and of spaces can be modified as desired.

F) - Characteristics and formats for input. Magnetic tape #2 ARCH3, produced by initial program is used as input. Its characteristics are:
7 tracks
800 BPI
Record length = 4 words
Block size = 255 records.
Label = ARCH3, binary.

Data entered from records refers to:
Sample number       ID       Integer
Date                DAY      Integer
BEGIN hours         HOUR     Integer
BEGIN minutes       MIN      Integer
End hours           NDHR     Integer
End minutes         NDMIN    Integer
Space code          ISPAC    Integer

According to the following format:

<table>
<thead>
<tr>
<th>ID</th>
<th>DAY</th>
<th>HOUR</th>
<th>MIN</th>
<th>NDHR</th>
<th>NDMIN</th>
<th>ISPAC</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>9</td>
</tr>
</tbody>
</table>

4 WODRS

DATA ENTERED THIS PROGRAM
Data entered from punched cards is used for headings and titles purposes, and is considered irrelevant for the process itself.

Data input by DATA statements is entered according to the following format:

DATA DOTS/9*1H,"./,X/"X"/
DATA NSPAC/26/

Data input as NAMELIST statements has the following format:

NAMELIST/CONST/BIGTOT,NTOTSD,TOTSD/AGUS/TOTSP,TOTDY,K
NAMELIST/CARM/CNT

$CONS BIGTOT=0.0,NTOTSD=182*0,TOTSD=182*0.0$
$AGUS TOTSP=26*0.0,TOTDY=7*0.0,K=0$
$CARM CNT=4368*0.0$

G)- Characteristics and formats for output. Output is handled by the line printer in two different ways:

a) Bar graphs
b) Summary table

Bar graphs contain:

Hour of the day, man-hours at that particular hour, and the graph itself, the reason for this format is a quick reference between actual man-hour figure and the bar representing it.

<table>
<thead>
<tr>
<th>Hour of the day</th>
<th>XTIME</th>
<th>Real</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of man-hours</td>
<td>CNT</td>
<td>Real</td>
</tr>
<tr>
<td>Graphical representation of man-hours</td>
<td>XLINE</td>
<td>Alpha</td>
</tr>
</tbody>
</table>
According to the following format:

Summary table contains:
Space, total man-hours, percent with respect to activity, percent with respect to day, for each day; and a final column with total man-hours per activity.

<table>
<thead>
<tr>
<th>Space name</th>
<th>SPCNAM</th>
<th>Alpha</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total man hours per space-day</td>
<td>NTOTSD</td>
<td>Integer</td>
</tr>
<tr>
<td>Percent of man-hours relative to space</td>
<td>PCTSP</td>
<td>Real</td>
</tr>
<tr>
<td>Percent of man-hours relative to day</td>
<td>PCTDY</td>
<td>Real</td>
</tr>
<tr>
<td>Total of man-hours per space</td>
<td>TOTSP</td>
<td>Real</td>
</tr>
</tbody>
</table>

According to the following format:
SAMPLE OF TABLE FORMAT

<table>
<thead>
<tr>
<th>SPACE</th>
<th>MONDAY</th>
<th>FRI</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPCNAM</td>
<td>NTOTSD(1)</td>
<td>PCTSP</td>
<td>PCTDY(1)</td>
</tr>
<tr>
<td>1</td>
<td>12</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>

I) - Input/Output designators:

Input : FILE 3, File name : ARCH3, Unit: TAPE
       FILE 5, File name : ARCH5, Unit: CARD READER

Output : FILE 6, File name : GORDA, Unit: LINE PRINTER

J) - Error messages. Checks for begin and end minutes values. Data outside of range are considered wrong and are printed for reference.

FMHDIST (III)

A) - Name of the program : FMHDIST/ARCH6

Language : Fortran

Estimated process time : 3.1 minutes
Compilation time : 14 secs.
Estimated core storage : 5888 words.

The program is used to read data from tape #2 ARCH3 about sample number, activities, days, beginning and ending time, and space, processing data and producing information about amounts of man-hours per space, day and activity, shows "loads" in terms of man-hours in spaces every day and according to each of the activities performed.

Information is presented in tables and summary tables, in two different ways:

a) Per activity, showing space vs. day
b) Per day, showing space vs. activity

B)- Solution method and procedures. A three-dimensional array CNT, with parameters space, activity, day, is the central element of the program. It is used to store values of man-hours related to each space, activity and day.

Procedure. Read in data from tape #2, ARCH3, concerning information from students.

Check validity of data, testing begin time, and end time, within range of 0-60 minutes, 0-24 for hours and equality of begin and end times. Invalid data is printed for reference.

Two temporary locations for time space are set: T1 for time relating hours and T2 for time relating minutes; begin and end times are compared and according to the logical results
the program branches into one of three different blocks, where time for minutes is computed. Both times for hours and for minutes determine the total time involved in each activity and this value is stored at CNT array, according to each space activity and day.

A series of auxiliary arrays stores total values to be used in summary table. Process recycles until input tape is exhausted and all arrays contain final figures. At this point a series of cycles carries out the process of computing final totals and producing tables.

C) - Assumptions. Information about time is by 5 minutes, range of values for hours 0-2400, range of value for minutes 0-60, range of days 1-7, range of activities 1-11, and range of spaces 1-26.

D) - Limitations. Input is limited by the format provided by input tape #2, ARCH3.

E) - User's options. The number of activities and spaces can be modified as required for each particular case.

F) - Characteristics and formats for input. Magnetic tape #2, ARCH3 is used as input, its characteristics are:

7 tracks
800 BPI
Record length = 4 words
Block size = 255 records.
Label = ARCH3, binary

Data entered from records refers to:

Sample number: ID (Integer)
Activity code: IACT (Integer)
Date: DAY (Integer)
Beginning hours: HOUR (Integer)
Beginning minutes: MIN (Integer)
Ending hours: NDHR (Integer)
Ending minutes: NDMIN (Integer)
Space code: ISPAC (Integer)

According to the following format:

![Format Diagram]

Data entered from punched cards is used for headings and titles purposes, and is considered irrelevant for the process itself.

Data input by DATA statements has the following format:

```
DATA CNT, TOTSD, TOTAD, TOTSA/2002*0., 182*0., 77*0., 286*0./
```
D) Characteristics and formats for output. Output is handled by the line printer in two different ways:
a) Tables per activities, showing space vs. days
b) Tables per days, showing space vs. activity

Tables per activities contain:
Space, total man-hours per space, activity and day, and total man-hours.

<table>
<thead>
<tr>
<th>Space name</th>
<th>SPCNAM</th>
<th>Alpha</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOTAD</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

According to the following format:
Summary table contains:
Space, total man-hours per space and day, and total man-hours per space:
- Space name: SPCNAM, Alpha
- Total man-hours per space-day: TOTSD, Real
- Total man-hours per space: TOTS, Real
- Total man-hours per day: TOTD, Real

According to the following format:
SAMPLE OF SUMMARY TABLE FORMAT

Tables per day contain:
Space, total man-hours per space, activity and day, and total man-hours.

Space name        SPCNAM       Alpha
Total man-hours per space, activity and day        CNT        Real
Total man-hours per space-activity        TOT        Real
Total man-hours per day        TOTAD        Real

According to the following format:
### Sample of Table Format

Summary table contains:

Space, total man-hours per space-activity, total man-hours per space.

<table>
<thead>
<tr>
<th>Space name</th>
<th>SPACNAM</th>
<th>Alpha</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total man-hours per space-activity</td>
<td>TOTSA</td>
<td>Real</td>
</tr>
<tr>
<td>Total man-hours per space</td>
<td>TOTS</td>
<td>Real</td>
</tr>
<tr>
<td>Total man-hours per activity</td>
<td>TOTA</td>
<td>Real</td>
</tr>
</tbody>
</table>

According to the following format:
**SAMPLE OF SUMMARY TABLE FORMAT**

I)- Input/Output designators.

**Input**: FILE 3, Name of file : ARCH3, Unit: TAPE  
FILE 5, Name of file : ARCH6, Unit: CARD READER  

**Output**: FILE 6, Name of file : SNOOPY, Unit: LINE PRINTER

J)- Error messages. When an error condition is produced by invalid data the record is printed for reference.

**Structural problem**

A)- Name of the program : STEEL/STRUCT.

Language : Fortran  
Estimated time : 0.9 minutes  
Compilation time : 39 secs.  
Estimated core storage : 4992 words.

The program is used to read two different kinds of data from cards referring to:
a) Variable input. Those data about conditions to be followed or satisfied in each particular situation.
b) Non-variable input. Those data about properties of sections that remain fixed for changing conditions of (a), but can be eventually modified, changed or extended according to user's needs.

Results are displayed in tables, containing information about the final sections and costs per square foot. Different alternatives for graphical display are explored.

Computations are carried out in order to release information about sections chosen, weight of steel used and final costs.

B)- Solution method and procedure. Program handles two main cases:
a) Beams spanning length and joists spanning width
b) Beams spanning width and joists spanning length
For both cases the process has two steps:
 i) Joist computations.
 ii) Beam computations.

For step i), process starts with load estimation on the joist, next computing shear, moment and deflection, according to:

\[ \text{SHEAR} = \frac{W \times L}{2} \]  (i.1)

Where : \( W \) = total load; units: pounds per linear foot.
\( L \) = span ; units: feet.
\( \text{shear} \) ; units: pounds.
MOMENT = \( \frac{W \times L^2}{8} \)  

Where:  
\( W \) = total load; units: pounds per linear foot.  
\( L^2 \) = span\(^2 \); units: square feet.  
MOMENT; units: pound-feet.  
DEFLECTION = \( \frac{5.0 \times W \times L^4 \times 1728}{384.0 \times E \times I} \)  

Where:  
\( W \) = total load; units: pounds per linear foot.  
\( E \) = Modulus of elasticity, units: pounds per square inch.  
\( I \) = Moment of inertia; units: inches\(^4 \)

When computations about joists are ready, computations about beams begin determining first the load on the beam and computing shear, moment and deflection, according to:

\[ \text{SHEAR} = \frac{W}{2} \]  

Where:  
\( W \) = total load; units: pounds shear; units: pounds  
MOMENT = \( \frac{W \times L}{8} \)  

Where:  
\( L \) = span; units: feet moment; units: pound-feet  
SECTIONS MODULUS: \( \frac{\text{MOMENT} \times 12}{\text{FALLB} \times 1000} \)  

Where:  
FALLB = Allowable bending stress; units: kips per square inch.  
section modulus; units: inches\(^3 \)

\[ \text{DEFLECTION} = \frac{5.0 \times W \times L^3 \times 1728}{384.0 \times E \times I} \]  

(ii.4)
Where \( E = \) modulus of elasticity; units: pounds per square inch.

\( I = \) moment of inertia ; units: inches exp.4

When joist and beam are selected or failure is reported, weights and costs per square foot are determined.

Process recycles for increments of 2 feet in length, until bay size of 40 X 40 feet is reached (i.e. 120 different conditions for each case)

At the end of that process the program calls one of a series of different subroutines that plot results in different form, to suit the user's requirements:

a) - Horizontal graph (line printer)
b) - Vertical graph (line printer)
c) - Horizontal graph (pen plotter)

This program was also worked in time-sharing in order to explore:

a) - a continuous vertical graph
b) - a situation in which, by means of supressing all output, only particular cases are analyzed and printed.

Procedure. A set of logical switches is defined, which take care of the different courses of action in the program, as follows:

(K) Computations for cases a and b have the following characteristics:
a)- Beams span length and joists span width. In this case each increment of length, up to 40 feet, adds only an extra joist, which characteristics are essentially the same.

b)- Beams span width and joists span length. In this case each increment of length, up to reaching 40 feet, adds load on the beam, even though it spans the same distance. Switch (K) takes care of this situation by controlling the program to compute only square bay joists, for case a saving computation time, for case b, joists and beams are computed each cycle.

(DEF and CARGA). These switches are used in association with one another and control whether user is using deflection of joists/beams as decision variable or not, and in any case, whether user considers total load or live load for computing deflection.

(AUX). Controls reports about failure conditions and their associated diagnosis.

(A and B). According to characteristics of cases a and b, two situations can happen: for (a), joist span is increased by 2 feet every time, beam span covers a complete cycle from square bay up to 40 feet. In this case, once a joist fails there are no further results since for the next computations joists will also fail. Nevertheless, a failure in a beam for some span will produce failure in next beams only up to the end of the cycle, where span reduces for next square bays,
thus, beam failure will not necessarily make all beams fail. The same concepts hold, reversed for case b, and the functions of A and B are to control the program in such a way that unnecessary computations (when a failure condition occurs) are skipped, saving computer time.

Other variables, though not logical in type, are used for logical control of the process. These are:

Joist's decision variable : ATEST
Beam's decision variable : BTEST

Variable DTEST takes on control of the weight comparison process, in order to ensure that weight comparisons for lightest sections is properly carried out.

Data are entered into the program by three different procedures:
a) Normal READ statements
b) DATA statements
c) NAMELIST statements

Procedures b and c cover all variable input and a carries out all non-variable input in order to facilitate the handling of the different alternatives to be studied.

Data entered and stored are printed for reference, with final results; and process starts setting three main cycles:
1) Cycle for spanning method. Covers cases a and b.
2) Cycle for width control. Covers the range of values that width can take on, from 12 to 40 feet in 2 feet increments.
Methods of computation involve calling length to joist span and width to beam span. The choice is arbitrary since both length and width are interchanged, depending upon actual spanning methods. Check is made for $K$ and cycle number for skipping unnecessary computation, also $A$ and/or $B$, for skipping computations.

Joist computation is carried out according to solution methods presented above in an iterative fashion, assigning first an estimated weight to the section and confirming or rejecting the choice. When a section is found a tolerance of 20% in weights (estimated and actual) is assumed. Along this process, ATEST keeps track of the decision variable governing the choice for further reference, when printed. When finally a satisfactory section (the lightest possible) is found, its characteristics are stored at: shape of joist: SHJ

weight of joist: WTJ

Beam computation continues checking whether available sections do exist or computations must be skipped. The process followed for beam selection is essentially the same as for joists.

In each case, the order of selection begins by selecting only from sections whose allowable span is consistent with actual span, avoiding also searching for sections which will not be used because of excessive span. Beam final choice is stored at shape of beam : SHB
weight of beam : WTB

At the end of this process, weights and costs per square foot are computed and results printed in tables. Total cost figures are stored in two arrays: TC1 and TC2.

When tables are printed the program searches for maximum and minimum cost (MAX, MIN).

These values are used to be plotted in graphs which display cost per square foot vs. bay size. Maximum and minimum values represent upper and lower limits in the range of costs. i.e., represent the size of the graph on the axis of costs.

A maximum value for this axis length must be provided because of the limitations of output devices. In order to bring the length of axis to an allowed size, this maximum value (MXVERT) is compared with actual range of cost values, and an index is computed when the range exceeds the allowed dimension (for cases in which actual range is less than allowed dimension, the index is set = 1).

According to such index all values of cost are normalized for use in graphs. At this point several subroutines can be called in order to plot the results, according to user's needs and equipment available.

1) Subroutine CARMEN. Used to plot horizontal graph through the line printer.
2) Subroutine LORENA. Used to plot vertical graph through the line printer.

3) Subroutine CCPLOT. Used to plot horizontal graph through the pen plotter. (drum plotter Calcomp, 11")

1) Subroutine CARMEN. Procedure: Subprogram receives information about:

Values stored in arrays TC1, TC2.

Maximum length of cost axis MXVERT.

Number of cases computed KK

Array to be plotted LINE

Special characters DOT, BLANK, STAR, ITAL, ZER.

Initial and final bay sizes L1, L2.

Joist spacing DIST.

Size of arrays II, JJ.

Maximum cost, minimum cost MAX, MIN.

Scale factor INDEX.

process starts normalizing MAX and MIN according to INDEX,
taking the value of MAX normalized, and beginning two cycles:

i) Cycle for equal cost search.

ii) Cycle for comparison of all bay sizes.

by means of which a search is made for values of both arrays which match with the actual cost determined, and place corresponding symbols in the graph for:

Case (a) : (X)

Case (b) : (H)

Equal cost, square bays : (0)
2) Subroutine LORENA. Procedure: Subprogram receives the same information as CARMEN, and performs the same normalizing actions, in order to determine relative values of cost when ranges exceed allowed dimension. A group of variables, ID, NI, IV and IVER, are used in order to control the changes in bay span variation, and keep track of square bay sizes.

A cycle for all bay sizes starts, and two temporary locations store values of KCC and KPC, representing upper and lower values of cost, for each bay size, according to cases (a) and (b), starting a new cycle ranging within these values in order to print a bar representing range of cost. Such bar is printed in two different ways:

i) When cost of method (a) exceeds that of (b) (X)
ii) When cost of method (b) exceeds that of (a) (H)
iii) Square bays or equal costs (0)

3) Subroutine CCPLOT. Procedure: Subprogram receives information about: X-coordinates array XCORD. Special symbols to be plotted VSCALE, SYMB, HSCALE, TITL1, TITL2, TITL3, IBUF, N1, N2, N3, N4, N5, IFIG, RTC1, RTC2 and all the information already described entered to CARMEN.

Process starts by setting the increments at which values and scale references will be plotted, in this case 0.20", (5 points per inch) is chosen, and all values of X and Y coordinates are normalized to that particular increment, determining the logical origin for the graph.
The process continues by drawing: vertical axis, values and title; perimeter of the graph; horizontal axis, values and vertical reference lines. After which the plot of values is carried out for each width increment.

A series of subroutines employed for this particular subprogram are provided by the system.

C)- Assumptions. 33 joists H-Series are considered; 56 different types of beams. Maximum length of cost scale:
52 lines, horizontal graph.
110 columns, vertical graph.
55 columns, vertical graph.*
45 units of 0.20", plotter.

Cases studied:
Joists and beams for roof and floor.
Deflection factor 360.0
Dead load 50.0 pounds per sq.ft.
Live load 30.0 pounds per sq. ft.
Estimated weight of joists 8.0 pounds per foot.
Estimated weight of beams 50.0 pounds per foot.
Steel A - 36.
Cost of steel in joists : 0.14 $ per pound.

* For graph produced through time-sharing terminal.
Cost of steel in beams 0.20 $ per pound.
Deflection is used, with total load in case (1).
Joist spacing 2.0 feet.
Minimum square bay 12.0 X 12.0 feet.
Maximum square bay 40.0 X 40.0 feet.

Structural elements are considered simply supported in all cases studied.

D) Limitations. No other considerations about support are provided. The program is limited to vary only within the support limits. The variable to be optimized is cost per square foot, but no consideration is made for depth, which can be included as another parameter.

E) User's options. User can modify:
Cases for floors/roofs to roofs only (ICASE)
Deflection factor.
Selection of deflection as decision variable
Consideration for load, determining the value of deflection.
Steel specifications.
Means for handling program, since in this case it was handled by tape, disk, and cards.

In input, options are: changing the means of representing graphical results, according to needs and capabilities.

One of the illustrations is a graph produced by pen plotter, which has the advantage of displaying the information in a
very complete form.

F)- Characteristics and formats in input. Data is entered completely by cards. Since the amount of data is small, this is convenient. Card input is considered in three forms: Normal READ statements; DATA statements; NAMELISTS statements.

For cases where program is transferred to disk for T.S., characteristics of disk are:
Record length = 10
Block size = 30
Serial.
Also for Calcomp plotter:
Record length = 1001
Block size = undefined.
556 BPI
Unlabeled.

Data entered from card records refer to all characteristics of joists and beams:

Allowed span for joists ALLJ Real
Joist section SHAPJ Alpha
Joist maximum allowed moment JMAXM Real
Joist maximum allowed shear JMAXS Real
Weight of joist ACTWTJ Real
Joist's moment of inertia INJ Real
Beam section SHAPB Alpha
Weight of beam \( \text{ACTWB} \) Real
Section modulus \( \text{SEC} \) Real
Beam's moment of inertia \( \text{INB} \) Real
Effective depth \( D \) Real
Webb thickness \( T \) Real
Allowed span \( \text{ALLB} \) Real
Bending stress code \( M \) Integer

According to the following format:

```
1 67 1617 2627 3637 4647 5657 6667 6869 80
1 67 1617 2627 3637 4647 5657 6667 6869 80
1 910 1819 2728 3637 4546 5455 6364 7273 80
... ... ... ... ... ... ... ... ...
1 67 1617 2627 3637 4647 5657 6667 6869 80

DATA CARDS SAMPLES
Data input by DATA statements is entered according to the following formats:

DATA BLANK, DOT, STAR, ITAL, ZER/1H , ",", "X", "H", "0"/
DATA L1, L2, DIST/12, 40, 2.0/
DATA NJO, NBM/33, 56/
DATA MXVERT/52/
DATA ICASE, FACTOR/2, 360.0/
DATA DEF, CARGA/. FALSE., TRUE./
DATA SP, MO, SH, DE, SM/"SPAN", "MOMENT", "SHEAR", "DEFLEC", "SH-MOM"/
DATA NO/"NONE",/FJ, FB/"J-FAIL", "B-FAIL"/
DATA II, JJ/120, 120/

Data input by NAMELIST statements is entered according to the following formats:

NAMELIST/CARGAS/LLOAD, DLOAD, ESTJ, ESTB/PROP/EVAL, FALLB, FALLV
NAMELIST/COSTOS/PRICEJ, PRICEB/REGINO/TC1, TC2
$CARGAS LLOAD=50.0, DLOAD=30.0, ESTJ=8.0, ESTB=50.0$
$PROP EVAL=2900000.0, FALLB(1)=24.0, FALLB(2)=22.0, FALLV=14500.0$
$COSTOS PRICEJ=14.0, PRICEB=20.0$
$REGINO TC1=120*0, TC2=120*0$

G)- Characteristics and formats for output. Output is presented in two different forms:

i)- Tables showing the sections chosen, diagnostic reports and costs.

ii)- Graphs (already discussed)
Tables contain:

Section choice, weight, deflections and costs, as well as decision variables:

- **Width**: \( W \) (Integer)
- **Length**: \( L \) (Integer)
- **Beam section**: \( SHB \) (Alpha)
- **Weight of beam section**: \( WTB \) (Real)
- **Actual deflection**: \( DELTB \) (Real)
- **Allowed deflection**: \( DELTXB \) (Real)
- **Beam weight per square foot**: \( TOTB \) (Real)
- **Beam cost per square foot**: \( COSTB \) (Real)
- **Decision variable**: \( ATEST \) (Alpha)
- **Joist section**: \( SHJ \) (Alpha)
- **Weight of joist section**: \( WTJ \) (Real)
- **Actual deflection**: \( DELTJ \) (Real)
- **Allowed deflection**: \( DELTXJ \) (Real)
- **Joist weight per square foot**: \( TOTJ \) (Real)
- **Joist cost per square foot**: \( COSTJ \) (Real)
- **Decision variable**: \( BTEST \) (Alpha)
- **Total cost per square foot**: \( TCOST \) (Real)

According to the following format:
I)- Input/Output designators.

Input  : FILE 5, File name : STRUCT, Unit : CARD READER
Output : FILE 6, File name : RATON, Unit : LINE PRINTER
Input  : FILE 5, File name : LORENA, Unit : DISK
Output : FILE 6, File name : RATON, Unit : REMOTE
         FILE 7, File name : CALTAP, Unit : TAPE.

For all listings of programs here described, as well as flow charts and samples of output, the reader is referred to the appendices, where the complementary documentation is presented.
A digression is made in order to call attention into another field, apparently different from that of architecture, and called Operation Research.

It is neither intended to make a detailed description of it, nor to deal deeply with the subject, which in itself deserves a complete work. The interested reader is referred to the bibliography for special treatises about O.R.\textsuperscript{14}

The purpose of the author, then, is to emphasize some aspects of O.R. from which he can draw experience, as another means to support his arguments.

It has to be kept in mind that no direct translation can be made from O.R. to architecture, but a reasoned analysis and a thorough understanding of the main features of O.R. will undoubtedly improve our ways for a better understanding of the architectural process.

O.R. is only new in name, and dates back to the forties, during World War II, in England, as a strong need for allocating scarce resources in an effective manner. Later the application went into industry and other fields, where its validity has been largely confirmed, e.g., aerospace problems,

\textsuperscript{14} The author rests mostly in Hillier and Lieberman; Churchman, Ackoff and Arnoff, Sasieni, Ackoff and Aronofsky, for the concepts stated here about O.R.
petroleum, communication, computer technology, electronics, transportation, distribution, governmental agencies, research programs, and hospitals, just to mention some fields where the application of O.R. has brought invaluable improvement.

Two main factors are considered as key elements in its rapid growth:

a) The progress made early in improving the techniques available to O.R.

b) The availability and constant improvement in the field of computers (Hillier and Lieberman). Special interest is placed in the following definition about O.R., as stated by Sasieni and Ackoff, who consider Operations Research as being:

"1) The application of scientific method
2) By interdisciplinary teams
3) To the problem-solving process of organized systems, so as to provide the best or optimal solution to the problems being considered", we can summarize this as defining it as the systems approach.

The application of scientific method in general is considered as being a process beginning with a careful identification of the problem, abstracting its essence, and the establishment of objectives (goals) to be fulfilled, in order to reach a proper definition of the problem. When this is achieved a model is derived, and the hypothesis that such a model is an
accurate representation of the real world problem, is made. This hypothesis is tested, verified, modified or rejected, during a feedback process.

As stated by Hillier and Lieberman, the phases of the scientific approach, in O.R. are:

"1)- Formulating the problem.
2)- Constructing a mathematical model to represent the system under study
3)- Deriving a solution from the model
4)- Testing the model and the solution derived from it.
5)- Establishing controls over the solution.
6)- Putting the solution to work: implementation"

It is of primary importance that the role played by the team, be an active one, not only by analyzing, but also by taking part in the decision-making process, resulting not only in improving actual situations, but rather in a constant effort to find optimal solutions.

This, of course, implies that no single man can embark upon such a task, and therefore the effective interaction of multidisciplinary teams is the only means to fulfill the objectives of O.R.

Once the problem has been identified and formulated and objectives (goals) are stated, a series of well-known methods are applied, according to the problem considered, for model
construction and testing.

At this point we can make a review relating to our primary interest, architecture. It becomes clear that problem definition/formulation, problem solving and solution implementation are processes in the sense that they are carried over a definite period of time in phases defined as programming, design and construction,¹⁵ and that architecture itself is a system in the sense that it is composed of a series of interacting components or subsystems.¹⁶

In this state of affairs we can establish a parallel between our work and that carried out by O.R. (a rough one perhaps, though no less relevant) because the application of some sort of scientific method to the process in architecture is essential in order to really understand the real nature of the problems to be solved.

That is an important objective since after all and in spite of the purely subjective-artistic oriented ones, it is of primary importance when making architecture, to provide optimal solutions in the broadest sense, i.e., not only formally

¹⁵. This assumption heavily rests upon "Problem Seeking" by W. Pena, CRS Publication, and the experience obtained by the author in a training period at CRS, in the programming department, headed also by Mr. W. Pena.

¹⁶. From here on when dealing with architecture we will use the term process and system, as defined here.
but also functionally and economically. And since there is a growth in number, kind and complexity in problems with more defined constraints, and most important, since our resources are not unlimited, it is the task of architects to provide a service that has to be directed to find such an optimal solution. (This is not an easy task, but perhaps reachable.)

In this sense then, the better formulated (understood) the problem, the better the solution; the better identified the problem, the more realistic and responsible the solution; and certainly it is the author's belief that we can expect to achieve these objectives through the application of a similar scientific approach to the whole process. The point to be made here, is that if computers have largely proved being the key tool in the O.R. process, there is no reason why their application can not also be a key tool for architecture.

No mention has been made of solution implementation, not because it is a less important part of the process, but because the author's aim is to emphasize that the "search for optimality" in the problem definition and problem solving parts of the process can be carried out precisely before solutions are implemented, consequently avoiding (or at least reducing to a minimum) the possibility that the final decisions are the wrong ones at a moment in which it is impossible to change them.
Obviously the importance of computers does not end there, since during the implementation part of the process they can be used also as key tools, as information analyzing devices for analysis and control of costs, scheduling, (computerized CPM applications are in fact a common practice), control of the construction process, formulation of specifications and their control, etc., just to mention a few applications.

As relating to aspect (2) of O.R., the parallel existing between the need for multidisciplinary teams in both O.R. and architecture is self-explanatory.

There exists no other possibility to cope with the complexity of problems faced by architectural practitioners. No one man can, as stated before, have such a comprehensive, superhuman capability for coping with so many things at a time, even if he is a "genius".

One thing is worthy of mention: it is the fact that a real multidisciplinary team can only be accomplished when improved communication systems are put to work, not only within, but in the outside relationship of the team. And again, at this point computers are an important part of such process, enabling communication of man to man, and man-machine itself, through processing and information retrieval tasks.

Finally, the third characteristic of O.R., i.e., the way in which optimality is sought, departs somehow from the same
task when related to architecture, particularly because the establishment of a mathematical model that properly reflects the real problem is not an easy or even clear task in architecture.

This point deserves a more detailed discussion, and is left to the next part.
Until this point the author has presented three major points:  
1) - The assumptions that lead to the statement of the thesis.  
2) - Examples which support such assumptions, and  
3) - The experience obtained and relationship of O.R. and Architecture.

This part intends to discuss more closely the nature of the man-machine interaction process.

No particular methods are suggested. Also, no claim for completeness is pretended. Those points to be dealt with from here on are simply the product of the three points made above, as the author's experience, and the point of departure for future development, improvement, extension or even rejection of his actual arguments.

It was mentioned in the last paragraph of part II that it is in the process of finding optimal solutions where O.R. and architecture depart from each other (which does not imply conflict), since it is easier for the O.R. process to develop a model that supposedly reflects the real world, a task which is quite difficult in architecture. The architecture system and its process are complex enough so as not to permit up to now their mathematical representation in a model which is also computationally feasible. In any event, it looks like we are still far from reaching such objective. However, this fact does not preclude that:
1) - Conceptually strong similarities do really exist between both fields, O.R. and architecture (which is the assumption already intended to be proved), and

2) - That we can apply properly available techniques, particularly when dealing with components, and eventually develop new ones, when dealing with the overall system. We must keep in mind that just as the sum of components does not imply the system itself, neither does optimality in components imply an overall optimization achievement.

Two questions arise at this point: one is are we going to be able to represent the system as a mathematical model, and the other, are we really supposed to direct all our efforts in order to achieve such a goal (in the event that it really is a goal).

It is the belief of the author that no final answer exists for the first question. Up to now this looks quite improbable, particularly about the subjective part of the process, but it is also true that applied mathematical sciences are developing rapidly and we certainly cannot foresee what these future developments will bring to us. This leads to the answer to question two, in the sense that as far as the author is concerned, we must direct our efforts to develop (with the whole body of team members) balanced but rational means of conducting the process, which help to better understand, state and solve problems, not only by means of the already traditional methods, but by applying other means, objectively and
subjectively feasible, so as to search for optimality.

Heuristic methods seem to the author promising means to achieve better statement and better solutions of problems.

Difficulty arises in the architectural process because:

1) - Conflicting situations between decision-makers (architect-client) teams, particularly at the outset; where computers can be an invaluable tool for preliminary analysis and communication means, so as to diminish these conflicts. No further steps can be effectively accomplished if no good care is taken at this point.

2) - Options are open, generally unknown in number and kind, and consequently not defined, particularly in new kinds of problems.

3) - As a result of 1) and 2), the impact produced by decisions is also unknown, and no precise measure of effectiveness exists (in the sense in which it can be stated in O.R.), however the definition of constraints, and the establishment of goals and concepts represent excellent tools for testing effects of decisions.

An again, computers play an important role by permitting team-members to store, retrieve, process and analyze vast amounts of information rapidly and accurately; examining which are the results of decisions, discarding bad ones, and exploring more detail those which promise improvement of the overall
system.

Let's then take a closer look at such man-machine processes by describing briefly first the system itself, related to man, and then the nature of such interaction.

Computers are considered to perform in two main forms: a) Open-loop systems, for which introduction of data does not affect the process; particularly used in information storage and retrieval (specifications, for example). b) Closed-loop systems, where introduction of data does affect the process. Used in problem definition, problem-solving processes.

Considering closed-loop systems, they are classified as: b1) Instruction generators, as used in industry (construction industry among others), when the computer controls mass-produced building subsystems. b2) Instruction executers. In this mode, it performs as it is instructed to do; this leads to the real man-machine interaction and we shall refer to it. Man-machine interaction can be performed in three main forms: 1) Direct access to the machine (principally when we are dealing with small machines). This implies high user efficiency and low machine efficiency. Here, the interface I/O units receive man instructions, transfer them to the

17. Davidson and Koenings.
18. This classification does not necessarily imply different systems; both can be carried out in the same machine.
computer, which in turn returns results to man.

2) Batch processing. This form became necessary as computer systems grew in size and complexity, making necessary the continuous processing of information in order to be profitable. This implies:

Low user efficiency and

High machine efficiency. Here, the user is not in direct control of the machine and therefore a very low level of communication does really exist.

Finally, the increase of speed, the development of mass-storage devices, and multiprocessing-multiprograming techniques gave rise to.

3) Time sharing, which implies real time process, from the user's concern, in a conversational mode.\(^\text{19}\)

Briefly then, those are the ways in which we can make use of computers, but how is this process carried out?\(^\text{20}\)

We can say that the process, as far as man-machine is concerned, can be carried out in two steps: (the order of mention does not imply hierarchy).

1) Components or subsystems.

\(^{19}\) Davidson and Koening.

\(^{20}\) The author heavily rests in an article published in *Progress in Operations Research.* (See Bibliography)
2) Interaction between them, which is the system itself.

Traditionally architects are capable of easily dealing with components of the systems, even when the process is carried out by other specialists. It is, however, his main task to deal with the interaction of such components, and it is here where, by traditional methods, i.e. the lack of a computer to help them, they have to rest only in their intuition and experience (assuming that they already have it), in the decision making process.

The computer here permits the practitioner to analyze efficiently (rapidly and accurately) the effects of decisions and to cope with the complexity of the problem.

(As stated before, and reemphasised here, computers do not miraculously bridge the gap, and ingenuity, subjectiveness and creativity are attributes which the team must possess.)

In order for this process to be properly carried out, we are required to know how to specify to the computer system the procedures we want to be performed, and from the computer to properly reference, retrieve, process and display information.

The better these requirements are met, the greater are the benefits to be obtained.

Essentially we have to be able to have complete access to the machine, so as to interact in a direct-immediate form and an-
alyze objectives and constraints, test and revise goals, to be confirmed or rejected, by looking upon the results of our decisions through feedback, in a flexible process.

This flexibility in man-machine communication is essential, and constitutes a primary goal to be achieved.

In part I it was said that computer applications must constitute interrelated-open entities, which can be used in any combination, according to the user's needs. If it is true that we have always to face very special cases, for which special "programs" have to be make, it is of paramount importance that more comprehensive means must be accomplished. A beginning has already been made. Problem-oriented languages, of which ICES and SIMSCRIPT, among many others can be mentioned, provide the software capability that enables such flexibility, not only for problem definition-solution, but also in better ways of handling great quantities and different kinds of data. It is implicit here that the computer, i.e., the hardware, has to provide also means to accomplish this, with increased speed and enough core, and auxiliary storage devices, as well as displaying properly information, particularly graphic display.

Such capabilities are relevant in the sense that they allow the user to:

a) Study the essence and uniqueness of his particular problem.
b) Interact efficiently with other members of the team, even when their personal ways of approaching the problem may differ. Also, the system when properly designed not only enables but also encourages creativity in the team members.

c) Play a definite role as decision maker in the whole process. This implies then a dynamic computer system, capable of processing and handling information and not only providing "numerical results".

Steven J. Fenves points out that "computers do not provide numbers, but merely insight into them". In the architectural field we must obtain meaningful measures of the consequences of our own decisions, thus broadening the scope of the whole architectural process, enabling users to make better decisions.

And assuming that architecture must be a response to its time, and ours being a computerized age, then architecture can find new expressions through this concept.

It has already been mentioned that man-machine interaction implies conversation with the computer, and that problem oriented languages, as opposed to procedure oriented languages, provide a logical means for such conversation. However, no integrated system as such for comprehensive architectural application has yet been achieved, though currently many applications directed to deal with components are in practice.
It seems a tremendous task to embark in the development of such a comprehensive goal, but there is no reason for not doing so, particularly if we think in the facilities of educational institutions.

In general it was said about problem oriented languages that they provide excellent communication means, and this is achieved because they are not "programmer" but "user" oriented, thus enabling the last to describe particular problems (essence) in particular situations (uniqueness), avoiding the painstaking task of writing a program for each problem, since they perform basic operations, in a modular fashion, rather than complete solutions.

This characteristic of modularity in P.O.L. allows the user to specify commands (as opposed to statements) that refer to complete sections of computer programs (composed of statements) and therefore can be used in any desired combination, and for any different problem.

Two different aspects of P.O.L. are worthy of mention:
1) - Procedure definition is separated from procedure specification, i.e., procedure specification is implied in the problem oriented commands.
2) - Data are no longer separated from the procedure, in fact they are intimately associated, and most important, input/output formats are not restricted and the user can choose, among several alternatives, the ones that better suit his
requirements.

This leads to better data handling concepts, particularly the capability of "data banks" which properly maintained, can be accessible at any time in varied forms by users.

So far, so good, but by what means can all of the above be applied, i.e., what can we say about the interface devices between man and machine?

These questions lead to the last aspect to be dealt with in this part, which are considered particularly important as far as the architect is concerned.

Input/Output Operations.

Graphical displays. These devices are a very important part of the computer system. They are of the most effectiveness as interface between man and machine, since graphics provide a general and rapid insight into the things they represent.

There are several ways for displaying graphic information:

a) The most common is the line-printer. It produces, together with usual results, graphical ones. It is however quite inflexible, and not very relevant, since it requires going to the whole process of specifying step by step to produce the desired results.
Next we have the pen plotters. These more flexible and very accurate and they are easy to operate since they provide both hardware - software facilities, and the process of application becomes a relatively simple one.

And finally, as the most important step up to now carried in the field we have the cathodic ray tubes with electronic pencils. This is the most flexible method of all, though also the most expensive one, at least until now. It permits the input/output process to be carried out instantly (storing and retrieving), permitting the comparison of different alternatives by superimposition, in the most natural form.

Furthermore, it permits modification and combination of such alternatives, allowing reference to the result of decisions, not only by one, but by many users at the same time, even in different places, a feature that no doubt increases the efficiency of the whole team.

Since this interface device permits a full conversation with the machine, the process of search for optimal solution can be carried out effectively, enabling a complete feedback process, and once final solutions are accomplished, these can be stored and eventually reproduced through microfilm or hard copies. This permits the storage of complete solutions as permanent files, which eventually become references for future problems.
Time-Sharing is taking its place by providing the user with the whole computer for his own use (in a conversational, real-time mode) in periods of time so short, that interruptions are imperceptible. It permits processes to be carried out in the most interacting way by providing immediate responses to different users, at different stages, giving total freedom for checking alternatives, and analyzing consequences of decisions without being thrown away each time an error, or a result are obtained. It avoids the problem of longer turn-arounds encountered in batch processing; the user is able to interact not only with the machine but with other users as well.

Combination of these devices, i.e., C.R.T., with time-sharing capabilities and multiprogramming with advanced software, will provide the architect's team with what they really need in order to carry out the whole process efficiently and fulfill the overall goal of providing the best possible solutions to any problem presented to them, in any kind of environment.
Summary.
The work dealt with the main aspect of the application of computers to the process of problem definition, problem solving and solution implementation.

In order to support the assumption that the computer system is by far the most powerful tool in this process, two examples were developed illustrating such application; reference was made also to the important effect of a different approach to the whole process by the architect.

It was emphasized that multidisciplinary teams are necessary as a result of that different attitude, and in order to cover all the aspects implied in the complexity of architectural problems.

Particular interest was placed upon the experience already obtained by Operations Research, and this led to the statement that along with the teams, a scientific approach is strongly needed.

Finally a review of man-machine interaction placed importance on the interface devices for Input/Output operations, and explored the impact of direct man-machine communication in terms of capability provided by multiprograming and Time-Sharing systems, and graphical displays of information.

It was pointed out that an efficient man-machine interaction
can only be achieved when man is in control of the process, and can "converse with the machine". Also, the importance of Problem Oriented Languages was mentioned.

All the aspects mentioned above, combined in a creative environment, can be used as the only means to obtain optimal solutions to the problems that are presented to architects.
Conclusions.
The conclusion is that in order to achieve the goal of computer applications, we have to consider:
1) That the computer as a rather new tool in our field implies a different attitude in approaching the problems, encouraging the development of multidisciplinary teams, applying scientific method, which in turn encourages creativity in the team members. This statement has already been proved in the Operations Research environment, from which a great deal of experience can be gained by architects.
2) That this experience cannot, and should not be transplanted directly to the architectural field but must be analyzed and properly understood, so that its useful features be considered when dealing with the proper means for coping with problems in the particular field of architecture.
3) That enough knowledge must be developed in order to really understand the man-machine interaction process, and to take advantage of the capabilities offered by actual and coming computer systems.
4) And finally and most important, that computer application, when properly carried out, requires an open mind, creative thinking, imagination, intuition, and experience in a balanced complementary fashion.

It is the author's hope that this be always kept in mind so
that our attitude is not to consider the computer as an evil, nor place it on a pedestal, so that man and computer can assume their proper roles in the whole process.
DMHDIST/ARCH2

TRANSFER DATA TO NEW TABLE IN APPROPRIATE FORM.
*CMH DIST / ARCH3*

**START**

1. **READ**

   **400**

   **LAST RECORD?**

   **T** → **4000**

   **F** → **200**

   **T** → **T**

   **F** → **F**

   **TIME IN**

   **PRINT**

   **COMPUTE**

   **VALUES FOR**

   **ACTIVITY, DAY AND**

   **HOURS**

   **END**

   **END**

**NOTE:** Block have been

numbeeed for further reference.
**NOTE:**

Blocks have been numbered for future reference.

```
START
READ IN K, N, Ttit, H, AND START CHARACTER
SET INITIAL VALUES CLEAR STORAGE LOCATIONS
COMPUTE INITIAL AND MEDIAN POINTS
COMPUTE CUMULATIVE INTERVALS AND IF FREQUENCIES
READ IN ON INPUT TAP
IF LAST RECORD
F
COMPUTE MEAN, VARIANCE, STD. ERROR AND LIMIT THE FREQUENCY
NORMALIZE CDF (ST. %) AND PDF, GRAPH pdf.
PRINT CDF
COMPUTE HISTOGRAM AND MODES
PRINT VALUES FOR ESTIMATED: MAX, MIN, MEAN, MEDIAN
END
```
A BLOCK NUMBER: 1

PART ONE

(FA)
C

K=1, N=1

XLINE (K) = X

PRINT A LINE OF DESIGN

XINP = X (IN/10)

IN=20

PRINT HEAD OF NEW PAGE

C00

C00

PRINT FOOT AND TITLES OF SUMMARY

TDO

ICT=1,8,30,35

TDO

L=1,7

TOTAL % EXCESS PER DAY
% PER DAY

TDO

PRINT TITLES PER DAY IN COLUMN

TDO

STOP
**ESTRUC/STEEL**

1. **INPUT FROM**
   - J035 & J036

2. **INPUT DATA**

3. **循环 for 批次**
   - **空间**

4. **循环 for 模拟**
   - **宽度**

5. **循环 for 个数**
   - **长度**

6. **循环 for 相位**

7. **BLK**: 1

8. **BLOCKS**: 2A, 2B

9. **BLOCKS**: 3A, 3B

10. **PRINT RESULTS**

11. **BLOCK 4**

12. **BLOCK 5**

13. **BLOCK 6**

* **CLICAL VARIABLES WILL DETERMINE**
  - * COURSE OF ACTION WHEN ONE ELEMENT FAILS*

*27/37*
**STRUC/STEEL**

**BLOCK 8 3A**

- Number of Stories (N)
- Assign Primary Weight to Beam
- Compute Area of Beam
- Load for Beam
- Shear = \( \frac{\text{Load}}{2} \)
- Moment, \( \frac{\text{Load} \times \text{Width}}{2} \)
- Section Modulus, \( \frac{\text{Moment}}{\text{Area of Beam}} \)

---

- Beam failing due to excessive weight
  - Choose cover plate by beam;
  - Assign secondary beam width
  - Diest 0

---

- Assign load
- Assign secondary weight

---

- Design 0

---

- Design 0

---

- Design 0
STRUCT/2500

VARIABLE

SUBROUTINE

BEGIN NO 4
*CCPLOT SUBROUTINE.*

$^*$ FOR CALCOMP PLOTTER.

($^*$ WIDTH)

**CCPLOT**

- Compute actual minimum and maximum costs.
- Set array for actual X-coord.
- Set array for actual Y-coord.
- Set array for actual Y-coord.

1. Define logical origin for graphplot.
2. Compute formal costs to define vertical axis.

3. Draw vertical axis.
   - Place formal cost scale.
   - Draw tittle of V. Axis.
   - Draw perimeter.

4. Draw horizontal axis.
   - Place formal size scale.
   - Draw tittle of H. Axis.
   - Draw lines for ref. square size.

5. Plot the graph of both spans on no meters, for each change in width.
   - Draw note of reference.

**RETURN**
CCPLOT SUBROUTINE

A FOR GRAPH PLOT ON
CALCOMP PLOTTER.

MIN = MIN/IND+.5
MAX = MAX/IND+.5

JK = 1,C2,(10)

A(JK) = (JK-1)/5.0
RBI(JK) = (DI(JK)-MIN)/5.0
RBI(JK) = (DI(JK)-MIN)/5.0

XI = -B10 3 XI = -1.00
Y1 = 60 3 Y2 = 0.05

BEGIN THE SEQUENCE
FOR FIXING LOGICAL
ORIGIN OF AXES.

J = 4, SCM

XMIN = (XMIN+90)*.5
COS90 = XMIN,.10.

PLOT VERTICAL
AXIS, AND V.
SCALE.

Y1 = Y1 + 0.20
Y2 = Y2 + 0.20
MIN = MIN+1

DRAW VERTICAL
AXIS, TITLE AND
PERIMETER.

A
CALL

DRAW SYMBOLS IN
NEXT SQUEEZE BAY
AND INITIALIZE MATRIX

N = N + ID

ID > 1

T

CALL

DRAW CODE FOR
SPANNING METHOD 1

CALL

DRAW CODE FOR
SPANNING METHOD 2

RETURN
FILE 10=UNIT(1),UNIT=TAPE,UNLABELED,ALPHA,BLOCKING=1,RECORD=900
FILE 12=UNIT(2),UNIT=TAPE,SAVE=999,BLOCKING=995,RECORD=1
FILE 6=UNIT(3),UNIT=PRINT
C THIS PROGRAM HAS BEEN WRITTEN TO USE THE EFL TAPE GENERATED.
C
C START BY SETTING ARRAYS TO STORE THE DATA CONTAINED
C IN ONE RECORD OF 900 WORDS FROM ORIGINAL TAPE.
C
DIMENSION ITEMP1(284),ITEMP2(284),ITEMP3(284),ITEMP4(284)
DIMENSION ITEMP5(284),ITEMP6(284),ITEMP7(284),ITEMP8(284)
DIMENSION ITEMP9(284)
IMC1=0
IMUS1=0
K = 0
C
READ ONE RECORD FROM THE ORIGINAL TAPE,
C AND PLACE THE DATA IN TEMPORARY ARRAYS.
C
5000 READ(10,1000,END=2000)ITEMP1(I),ITEMP2(I),ITEMP3(I),ITEMP4(I)
            ,ITEMP5(I),ITEMP6(I),ITEMP7(I),ITEMP8(I)
            ,ITEMP9(I),I=1,284
C
ASSIGN VALUES TO ACTUAL VARIABLES FROM DATA STORED
IN THE TEMPORARY ARRAYS.
C
20 IDV01 = ITEMP1(1)
         IDV1 = ITEMP2(1)
         IDV2 = ITEMP3(1)
         IDV3 = ITEMP4(1)
         IDV4 = ITEMP5(1)
         IDV5 = ITEMP6(1)
         IDV6 = ITEMP7(1)
         IDV7 = ITEMP8(1)
         IDV8 = ITEMP9(1)
C
WHEN THE SET JUST READ IS THE FIRST ONE IN THE FILE,
SAVE IT AND PICK UP A NEW SET, FOR COMPARISONS.
C
IF (K) 40,150,40
150 K = 1
GO TO 160
C
CHECK IF THE SAMPLE NUMBERS DO MATCH.
C
40 IF (IDV02 - IDV01) .GT. 50 .OR. 140
C IF THEY DO MATCH COME HERE AND CHECK IF DATES DO TOO,
C OTHERWISE MODIFY RECORD AND WRITE IT OUT.
C
50 IF (IDAT2 - IDAT1) 80,40,80
C IF THEY DO MATCH COME HERE AND WRITE OUT THE DATA,
C OTHERWISE MODIFY THE RECORD AND WRITE IT OUT.
C
60 IF (IDV1 = IDV61
IROOM2 = IROOM1
ISUFF2 = ISUFF1
ISPCT2 = ISPCT1
IHMUS2 = IHMUS1
I = I + 1
IF (1 = 200) 20*20, 5000

COME HERE WHEN THE END OF FILE IS REACHED IN INPUT TAPE.

2000 IEND = 2400
ISFQ = ISFQ + 1
WRITE (12, 1002) IDNO2, ISFQ, ISEC2, IHMUS2
IF (IDNO2 = LE. 3) WRITE (6, 1003) IDNO2, ISFQ, ISEC2, IHMUS2
ENDFILE 12
WRITE (6, 1004)

FORMAT STATEMENTS.

1000 FORMAT (2A8, 13*I, 12*1, 13*I)
1001 FORMAT (1X, "EXTRA CARD FOR CHANGE OF DAYS," 12*X, 13*I)
1002 FORMAT (13*I, 12*1, 13*I)
1003 FORMAT (1X, "REGULAR CARDS," 12*X, 13*I)
1004 FORMAT (1X, "END OF FILE REACHED IN INPUT TAPE"")
1005 FORMAT (1X, "EXTRA CARD FOR CHANGE OF SAMPLE," 12*X, 13*I)
200 FORMAT (1X, "SAMPLE")
2005 FORMAT (1X, "SAMPLE")
2015 FORMAT (1X, "SAMPLE")
2025 FORMAT (1X, "SAMPLE")
2035 FORMAT (1X, "SAMPLE")
2045 FORMAT (1X, "SAMPLE")
2055 FORMAT (1X, "SAMPLE")
2065 FORMAT (1X, "SAMPLE")
2075 FORMAT (1X, "SAMPLE")
2085 FORMAT (1X, "SAMPLE")
2095 FORMAT (1X, "SAMPLE")
300 FORMAT (1X, "SAMPLE")
3005 FORMAT (1X, "SAMPLE")
3015 FORMAT (1X, "SAMPLE")
3025 FORMAT (1X, "SAMPLE")
3035 FORMAT (1X, "SAMPLE")
3045 FORMAT (1X, "SAMPLE")
3055 FORMAT (1X, "SAMPLE")
3065 FORMAT (1X, "SAMPLE")
3075 FORMAT (1X, "SAMPLE")
3085 FORMAT (1X, "SAMPLE")
3095 FORMAT (1X, "SAMPLE")
400 FORMAT (1X, "SAMPLE")
4005 FORMAT (1X, "SAMPLE")
4015 FORMAT (1X, "SAMPLE")
4025 FORMAT (1X, "SAMPLE")
4035 FORMAT (1X, "SAMPLE")
4045 FORMAT (1X, "SAMPLE")
4055 FORMAT (1X, "SAMPLE")
4065 FORMAT (1X, "SAMPLE")
4075 FORMAT (1X, "SAMPLE")
4085 FORMAT (1X, "SAMPLE")
4095 FORMAT (1X, "SAMPLE")
500 FORMAT (1X, "SAMPLE")
5005 FORMAT (1X, "SAMPLE")
5015 FORMAT (1X, "SAMPLE")
5025 FORMAT (1X, "SAMPLE")
5035 FORMAT (1X, "SAMPLE")
5045 FORMAT (1X, "SAMPLE")
5055 FORMAT (1X, "SAMPLE")
5065 FORMAT (1X, "SAMPLE")
5075 FORMAT (1X, "SAMPLE")
5085 FORMAT (1X, "SAMPLE")
5095 FORMAT (1X, "SAMPLE")
600 FORMAT (1X, "SAMPLE")
6005 FORMAT (1X, "SAMPLE")
6015 FORMAT (1X, "SAMPLE")
6025 FORMAT (1X, "SAMPLE")
6035 FORMAT (1X, "SAMPLE")
6045 FORMAT (1X, "SAMPLE")
6055 FORMAT (1X, "SAMPLE")
6065 FORMAT (1X, "SAMPLE")
6075 FORMAT (1X, "SAMPLE")
6085 FORMAT (1X, "SAMPLE")
6095 FORMAT (1X, "SAMPLE")
700 FORMAT (1X, "SAMPLE")
7005 FORMAT (1X, "SAMPLE")
7015 FORMAT (1X, "SAMPLE")
7025 FORMAT (1X, "SAMPLE")
7035 FORMAT (1X, "SAMPLE")
7045 FORMAT (1X, "SAMPLE")
7055 FORMAT (1X, "SAMPLE")
7065 FORMAT (1X, "SAMPLE")
7075 FORMAT (1X, "SAMPLE")
7085 FORMAT (1X, "SAMPLE")
7095 FORMAT (1X, "SAMPLE")
800 FORMAT (1X, "SAMPLE")
8005 FORMAT (1X, "SAMPLE")
8015 FORMAT (1X, "SAMPLE")
8025 FORMAT (1X, "SAMPLE")
8035 FORMAT (1X, "SAMPLE")
8045 FORMAT (1X, "SAMPLE")
8055 FORMAT (1X, "SAMPLE")
8065 FORMAT (1X, "SAMPLE")
8075 FORMAT (1X, "SAMPLE")
8085 FORMAT (1X, "SAMPLE")
8095 FORMAT (1X, "SAMPLE")
900 FORMAT (1X, "SAMPLE")
9005 FORMAT (1X, "SAMPLE")
9015 FORMAT (1X, "SAMPLE")
9025 FORMAT (1X, "SAMPLE")
9035 FORMAT (1X, "SAMPLE")
9045 FORMAT (1X, "SAMPLE")
9055 FORMAT (1X, "SAMPLE")
9065 FORMAT (1X, "SAMPLE")
9075 FORMAT (1X, "SAMPLE")
9085 FORMAT (1X, "SAMPLE")
9095 FORMAT (1X, "SAMPLE")

SEGMENT 1 IS 488 LONG
FILE 2=ARCH1,UNIT=READER
FILE 6=PRINT,UNIT=PRINT
FILE 10=ARCH2,UNIT=TAPE,SAVE=999,BLOCKING=341,RECORD=3
C
C PROGRAM NAME " ANH01ST/ARCHI."
C INPUT SOURCE: DATA CARDS WITH SPACE CODE ADDED.
C OUTPUT: ARCH1 TAPE.
C DATA COLLECTED AT RICE UNIVERSITY FALL 1969-1970
C SAMPLE SIZE "33"
C SAMPLE DRAWN FROM FIRST AND FIFTH YEAR OF ARCHITECTURE.
C BEFORE GENERATING THE NEW TAPE, A CHECK IS MADE OF:
C SAMPLE NUMBER, SPECIFY DAY, CHECKING TIME, AND IF THEY ARE IN
C CHECK RANGE, THEN PRINT BACK ON TAPE.
C THIS WILL SCREEN THE INFORMATION, REFUSE PROCESSING.

START OF SEGMENT ********** 1

ISAMPL=43
NACTV=11
NSPAC=26

K=0

1000 READ(5,1000,END=99) IDND,IACTV,DATA,INBGT,INISZ,ISNUM
          00000170  R 0003
          00000140  R 0019
          00000190  R 0025
          00000200  R 0028
          00000210  R 0032
          00000220  R 0035
          00000230  R 0039
          00000240  R 0043
          00000250  R 0057
          00000260  R 0064
          00000270  R 0067
          00000280  R 0072
          00000290  R 0079
          00000300  R 0087
          00000310  R 0114
          00000320  R 0114

1000 IF (IDND.LE.0,OR,IDND.GT.ISAMPLE) GO TO 100
      IF (IACTV.LE.0,OR,IACTV.GT.NACTV) GO TO 100
      IF (DATA.LE.0,OR,DAT.GT.NACTV) GO TO 100
      IF (INBGT.GT.2000) GO TO 100
      IF (INISZ.GT.ISPAC) GO TO 100
          IF (INNUM.EQ.04) WRITE(5,1002) IDND,IACTV,DATA,INBGT,
          INISZ,ISNUM,ISPAC
          00000240  R 0043
          00000250  R 0057
          00000260  R 0064
          00000270  R 0067
          00000280  R 0072
          00000290  R 0079
          00000300  R 0087
          00000310  R 0114

1000 IF (IDND.LE.0,OR,IDND.GT.ISAMPLE) GO TO 100
      IF (IACTV.LE.0,OR,IACTV.GT.NACTV) GO TO 100
      IF (DATA.LE.0,OR,DAT.GT.NACTV) GO TO 100
      IF (INBGT.GT.2000) GO TO 100
      IF (INISZ.GT.ISPAC) GO TO 100
          IF (INNUM.EQ.04) WRITE(5,1002) IDND,IACTV,DATA,INBGT,
          INISZ,ISNUM,ISPAC
          00000240  R 0043
          00000250  R 0057
          00000260  R 0064
          00000270  R 0067
          00000280  R 0072
          00000290  R 0079
          00000300  R 0087
          00000310  R 0114

C FORUHAT STATEMENTS.
C 1000 FORUHAT (13+1X+12+1X+12+1X+13+1X+13+1X+13+1X+13)
      00000320  R 0116
      00000330  R 0116
      00000340  R 0116
      00000350  R 0116
      00000360  R 0116
      00000370  R 0116
      00000380  R 0116
      00000390  R 0116
      00000400  R 0116
      00000410  R 0129

END

SEGMENT 1 IS 138 LONG
<table>
<thead>
<tr>
<th>PROCESS TIME</th>
<th>5.79 MIN.</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>OK RECORD</th>
<th>4</th>
<th>6</th>
<th>1230</th>
<th>2</th>
<th>50</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>OK RECORD</td>
<td>4</td>
<td>9</td>
<td>1250</td>
<td>1</td>
<td>1</td>
<td>25</td>
</tr>
<tr>
<td>OK RECORD</td>
<td>4</td>
<td>3</td>
<td>1300</td>
<td>2</td>
<td>60</td>
<td>2</td>
</tr>
<tr>
<td>OK RECORD</td>
<td>4</td>
<td>9</td>
<td>1370</td>
<td>2</td>
<td>24</td>
<td>2</td>
</tr>
<tr>
<td>OK RECORD</td>
<td>4</td>
<td>6</td>
<td>1800</td>
<td>2</td>
<td>200</td>
<td>1</td>
</tr>
<tr>
<td>OK RECORD</td>
<td>4</td>
<td>10</td>
<td>1830</td>
<td>2</td>
<td>24</td>
<td>2</td>
</tr>
<tr>
<td>OK RECORD</td>
<td>4</td>
<td>8</td>
<td>1880</td>
<td>1</td>
<td>1</td>
<td>25</td>
</tr>
<tr>
<td>OK RECORD</td>
<td>4</td>
<td>4</td>
<td>2000</td>
<td>1</td>
<td>1</td>
<td>25</td>
</tr>
<tr>
<td>OK RECORD</td>
<td>4</td>
<td>8</td>
<td>2100</td>
<td>2</td>
<td>24</td>
<td>2</td>
</tr>
<tr>
<td>OK RECORD</td>
<td>4</td>
<td>4</td>
<td>2130</td>
<td>1</td>
<td>24</td>
<td>2</td>
</tr>
<tr>
<td>OK RECORD</td>
<td>4</td>
<td>11</td>
<td>2230</td>
<td>2</td>
<td>24</td>
<td>2</td>
</tr>
<tr>
<td>OK RECORD</td>
<td>4</td>
<td>7</td>
<td>5</td>
<td>115</td>
<td>2</td>
<td>24</td>
</tr>
<tr>
<td>OK RECORD</td>
<td>4</td>
<td>8</td>
<td>5</td>
<td>200</td>
<td>1</td>
<td>24</td>
</tr>
<tr>
<td>OK RECORD</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>230</td>
<td>2</td>
<td>24</td>
</tr>
<tr>
<td>OK RECORD</td>
<td>4</td>
<td>7</td>
<td>5</td>
<td>795</td>
<td>1</td>
<td>24</td>
</tr>
<tr>
<td>OK RECORD</td>
<td>4</td>
<td>10</td>
<td>5</td>
<td>755</td>
<td>2</td>
<td>24</td>
</tr>
<tr>
<td>OK RECORD</td>
<td>4</td>
<td>3</td>
<td>5</td>
<td>805</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>OK RECORD</td>
<td>4</td>
<td>10</td>
<td>5</td>
<td>945</td>
<td>3</td>
<td>24</td>
</tr>
<tr>
<td>OK RECORD</td>
<td>4</td>
<td>15</td>
<td>955</td>
<td>1</td>
<td>1</td>
<td>25</td>
</tr>
<tr>
<td>OK RECORD</td>
<td>4</td>
<td>9</td>
<td>5</td>
<td>1055</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>OK RECORD</td>
<td>4</td>
<td>8</td>
<td>5</td>
<td>1115</td>
<td>2</td>
<td>24</td>
</tr>
<tr>
<td>OK RECORD</td>
<td>4</td>
<td>10</td>
<td>5</td>
<td>1200</td>
<td>2</td>
<td>24</td>
</tr>
<tr>
<td>OK RECORD</td>
<td>4</td>
<td>6</td>
<td>5</td>
<td>1210</td>
<td>2</td>
<td>150</td>
</tr>
<tr>
<td>OK RECORD</td>
<td>4</td>
<td>10</td>
<td>5</td>
<td>1230</td>
<td>2</td>
<td>24</td>
</tr>
<tr>
<td>OK RECORD</td>
<td>4</td>
<td>8</td>
<td>5</td>
<td>1300</td>
<td>2</td>
<td>40</td>
</tr>
<tr>
<td>OK RECORD</td>
<td>4</td>
<td>9</td>
<td>5</td>
<td>1370</td>
<td>2</td>
<td>24</td>
</tr>
<tr>
<td>OK RECORD</td>
<td>4</td>
<td>11</td>
<td>5</td>
<td>1740</td>
<td>2</td>
<td>24</td>
</tr>
<tr>
<td>OK RECORD</td>
<td>4</td>
<td>6</td>
<td>5</td>
<td>1800</td>
<td>2</td>
<td>200</td>
</tr>
<tr>
<td>OK RECORD</td>
<td>4</td>
<td>9</td>
<td>5</td>
<td>1835</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>OK RECORD</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>1900</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>OK RECORD</td>
<td>4</td>
<td>8</td>
<td>5</td>
<td>2030</td>
<td>2</td>
<td>24</td>
</tr>
<tr>
<td>OK RECORD</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>2900</td>
<td>1</td>
<td>24</td>
</tr>
<tr>
<td>OK RECORD</td>
<td>4</td>
<td>7</td>
<td>5</td>
<td>2330</td>
<td>2</td>
<td>24</td>
</tr>
<tr>
<td>OK RECORD</td>
<td>4</td>
<td>7</td>
<td>5</td>
<td>2400</td>
<td>2</td>
<td>24</td>
</tr>
</tbody>
</table>

### HAD RECORD

<table>
<thead>
<tr>
<th>HAD RECORD</th>
<th>1</th>
<th>0</th>
<th>0</th>
<th>103</th>
<th>0</th>
<th>0</th>
<th>2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>HAD RECORD</td>
<td>0</td>
<td>8</td>
<td>10</td>
<td>0</td>
<td>11</td>
<td>1</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>HAD RECORD</td>
<td>31</td>
<td>35</td>
<td>6</td>
<td>1500</td>
<td>1</td>
<td>2</td>
<td>25</td>
<td>3</td>
</tr>
<tr>
<td>HAD RECORD</td>
<td>31</td>
<td>4</td>
<td>11</td>
<td>1615</td>
<td>2</td>
<td>20</td>
<td>25</td>
<td>4</td>
</tr>
<tr>
<td>HAD RECORD</td>
<td>31</td>
<td>39</td>
<td>1</td>
<td>2130</td>
<td>3</td>
<td>10</td>
<td>24</td>
<td>5</td>
</tr>
<tr>
<td>HAD RECORD</td>
<td>39</td>
<td>11</td>
<td>1</td>
<td>2545</td>
<td>1</td>
<td>1</td>
<td>25</td>
<td>6</td>
</tr>
<tr>
<td>HAD RECORD</td>
<td>39</td>
<td>4</td>
<td>2</td>
<td>2430</td>
<td>1</td>
<td>1</td>
<td>25</td>
<td>7</td>
</tr>
<tr>
<td>HAD RECORD</td>
<td>39</td>
<td>5</td>
<td>3</td>
<td>2430</td>
<td>1</td>
<td>1</td>
<td>25</td>
<td>8</td>
</tr>
<tr>
<td>HAD RECORD</td>
<td>39</td>
<td>5</td>
<td>4</td>
<td>2430</td>
<td>1</td>
<td>2</td>
<td>25</td>
<td>9</td>
</tr>
<tr>
<td>HAD RECORD</td>
<td>40</td>
<td>20</td>
<td>7</td>
<td>230</td>
<td>2</td>
<td>25</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>HAD RECORD</td>
<td>40</td>
<td>22</td>
<td>2</td>
<td>743</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>11</td>
</tr>
<tr>
<td>HAD RECORD</td>
<td>40</td>
<td>22</td>
<td>2</td>
<td>840</td>
<td>1</td>
<td>1</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>HAD RECORD</td>
<td>40</td>
<td>22</td>
<td>2</td>
<td>1450</td>
<td>1</td>
<td>6</td>
<td>25</td>
<td>13</td>
</tr>
<tr>
<td>HAD RECORD</td>
<td>40</td>
<td>22</td>
<td>3</td>
<td>940</td>
<td>1</td>
<td>1</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>HAD RECORD</td>
<td>40</td>
<td>22</td>
<td>3</td>
<td>1145</td>
<td>4</td>
<td>10</td>
<td>1</td>
<td>15</td>
</tr>
<tr>
<td>HAD RECORD</td>
<td>34</td>
<td>6</td>
<td>3</td>
<td>1225</td>
<td>2</td>
<td>25</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>HAD RECORD</td>
<td>63</td>
<td>40</td>
<td>10</td>
<td>9050</td>
<td>10</td>
<td>10</td>
<td>25</td>
<td>17</td>
</tr>
</tbody>
</table>
START OF SEGMENT *********** 1

K = 0

READ IN FIRST RECORD FROM ARCH2 TAPE.

READ(I0,1000) INNP, TACTV1, IDAT1, TREG1, IGSIZ1, ISNUM1
  1 1SRO1C

SINCE THE SEQUENCE NUMBER NUMBERS WITHIN DAYS, ALWAYS
Q = QINIT TO ISEQ FOR CHANGE OF DAY AND SAMPLE NUMBER.

IF ISEQ = 0

SAVE OLD DATA INTO THOBS TO MAKE ROOM FOR NEW DATA

TO USE FOR COMPARISON.

IF IDATQ = IDATQ1
  TACTVQ = TACTVQ1
  IDATQ = IDATQ1
  IGSIZQ = IGSIZQ1
  ISNUMQ = ISNUMQ1
  1SROQ = 1SROQ1

READ ANOTHER RECORD FROM ARCH2 TAPE, AGAIN.

READ(I0,1000) INNP, TACTVQ, IDATQ, TREGQ, IGSIZQ, ISNUMQ, 1SROQ

CHECK IF THE SAMPLE NUMBERS DO MATCH.

IF THEY DO NOT MATCH, SAVE THE NEW RECORD

AND READ ANOTHER ONE FROM ARCH2 TAPE.

IF (1EQ0, EQ1, INNP1) GO TO 20

IF (ISEQ1 = ISEQQ) GO TO 21

WRITE (17, 1002) INNP2, TACTV2, IDAT2, ISEQ2, TREGQ, IGSIZQ
  1 1SROQ, 1SROQ2

WRITE OUT A SAMPLE RECORD.
IF (1ND2.GT.36) WRITE(6,1005)1ND2,IACTV,IData2,(SF3,16612)
  1. GO TO 10

2A K = K+1
WRITE(6,1006)1ND2,IACTV,IData2,IS6,16612,1ENDT,16672
  1. ISNUM2,ISPAC2
  GO TO 10

3A IF SAMPLE NUMBERS DO NOT MATCH, CHECK IF DATES DO TOO.

3B IF THE DATES DO NOT MATCH, GENERATE AN EXTRA CARD.

3C IF (1ND2.EQ.IDAT1) GO TO 50

3D CAME HERE WHEN SAMPLE NUMBERS AND DATES MATCH.

3E 1ENDT=16671
  IS6 = IS6+1
  IF (1ND2.EQ.16612) GO TO 45

4C WRITE OUT A RECORD WITH THE FORMAT NEEDED FOR THE
   PLOTTING ROUTINE.
   WRITE (12,1002) 1ND2,IACTV,IData2,IS6,16612,1ENDT,16672
  1. ISNUM2,ISPAC2

4D WRITE OUT A SAMPLE RECORD.
   1. IF (1ND2.EQ.16612) WRITE(6,1061)1ND2,IACTV,IData2,IS6,16612,1ENDT,16672
   1. ISNUM2,ISPAC2
   GO TO 20

4E CAME HERE WHEN AN EXTRA CARD MUST BE GENERATED.

50 1ENDT=2400
  IS6 = IS6+1
  IF (1ND2.EQ.16612) GO TO 52

5C WRITE OUT A RECORD IN THE FORMAT NEEDED FOR THE PLOTTING ROUTINE.
   WRITE (12,1002) 1ND2,IACTV,IData2,IS6,16612,1ENDT,16672
  1. ISNUM2,ISPAC2

5D WRITE OUT A SAMPLE RECORD.
   1. IF (1ND2.EQ.16612) WRITE(6,1003)1ND2,IACTV,IData2,IS6,16612,1ENDT,16672
   1. ISNUM2,ISPAC2
   GO TO 20

5E CAME HERE WHEN AN EXTRA CARD MUST BE GENERATED.

52 1ENDT=2400
  IS6 = IS6+1
  IF (1ND2.EQ.16612) GO TO 52

5C WRITE OUT A RECORD IN THE FORMAT NEEDED FOR THE PLOTTING ROUTINE.
   WRITE (12,1002) 1ND2,IACTV,IData2,IS6,16612,1ENDT,16672
  1. ISNUM2,ISPAC2

5D WRITE OUT A SAMPLE RECORD.
   1. IF (1ND2.EQ.16612) WRITE(6,1003)1ND2,IACTV,IData2,IS6,16612,1ENDT,16672
   1. ISNUM2,ISPAC2
   GO TO 20

5E CAME HERE WHEN AN EXTRA CARD MUST BE GENERATED.
WRITE OUT A RECORD IN THE FORMAT NEEDED FOR THE PLOTTING ROUTINE.

WRITE (12,1002) IUN, 10, IACTV2, IDAT2, ISEQ, IBEGIN, IEND, 1GSIZE
1 ISNUM2, ISPACE2
PRINT OUT A SAMPLE RECORD.
IF (IUN = 1) WRITE (12,1001) IUN, 10, IACTV2, IDAT2, ISEQ, IBEGIN, IEND, 1GSIZE
1 ISNUM2, ISPACE2
GO TO 20
50 K = K + 1
WRITE (12,1006) IUN, 10, IACTV2, IDAT2, ISEQ, IBEGIN, IEND, 1GSIZE
1 ISNUM2, ISPACE2
GO TO 10
56 ISIZE = 0
GO TO 20
COME HERE WHEN THE END OF FILE IS REACHED ON THE INPUT TAPE.
2000 ISIZE = ISIZE + 1
IUN = 2
IF (IEND = IBEGIN) GO TO 2005
WRITE OUT A RECORD IN THE FORMAT NEEDED FOR THE PLOTTING ROUTINE.
WRITE (12,1002) IUN, 10, IACTV2, IDAT2, ISEQ, IBEGIN, IEND, 1GSIZE
1 ISNUM2, ISPACE2
IF (IUN = 1) WRITE (12,1001) IUN, 10, IACTV2, IDAT2, ISEQ, IBEGIN, IEND, 1GSIZE
1 ISNUM2, ISPACE2
GO TO 2010
2005 K = K + 1
WRITE (12,1006) IUN, 10, IACTV2, IDAT2, ISEQ, IBEGIN, IEND, 1GSIZE
1 ISNUM2, ISPACE2
2010 END FILE 12
WRITE (12,1004) PTIME = TIME() / 3600
WRITE (12,1300) PTIME

FORMAT STATEMENTS
1000 FORMAT (15,12,17,14,13,17)
1001 FORMAT (15,'EXTRA CARD FOR CHANGE OF DAYS."
113,2X/(14,12,17,14,13,17)
1002 FORMAT (15,12,17,14,13,17)"
1003 FORMAT (15,'"REGULAR CAMPS."
113,2X/(14,12,17,14,13,17)"
1004 FORMAT (15,'"END OF FILE REACHED ON INPUT TAPE."
1005 FORMAT (15,'"CHANGE OF DAYS."
113,2X/(14,12,17,14,13,17)"
1006 FORMAT (15,'"END OF FILE REACHED ON INPUT TAPE."
113,2X/(14,12,17,14,13,17)"
1500 FORMAT (15,'"PROCESS TIME = ",X(12,2X,"MIN."
STOP
END
FILE 5=ARCH4, UNIT=READER
FILE 3=ARCH4, UNIT=TAPE, BLKINC=255, RECORD=8
FILE 6=PRINT, UNIT=PRINT

DIMENSION (T1(127), ACT1(11), DYT1(7), TOT(117))

REAL (T1(127), DAY*(T1), SCALE*(T1), XTIME*(T1), ACTN(11,2))

INILG PCTAC(7), PCTDY(7)
INILG DAYCD, HOUR
REINI 3

READ HEADER AND ALL STUFF TO BEGIN.

READ (>, 1000) HEAD
READ (>, 1001) (DAY(I,J), J=1, 2, I=1, 7)
READ (>, 1002) SCALE
READ (>, 1003) NYTS
READ (>, 1006) X
READ (>, 1005) XTIME
READ (>, 1007) (ACTN(I,J), J=1, 2, I=1, 11)

NEXT, LET 5 CLEAR STORAGE AREAS, TO BE SURE NOTHING IS THERE.

G11 = 0.0
DO 50 I=1, 11
   ACTN(I) = 0.0
   DO 50 J=1, 7
      T1(J) = 1.0
      DYT1(J) = 0.0
   DO 50 I=1, 11
50 CONTINUE

K11 = 1
   DO 150 I=1, 11
      J=1, 7
      D15 = 1.0
      COUNT(I,J,K) = 0.0
   DO 150 I=1, 11
150 CONTINUE

JACTY = 0
WHITE (6, 1011)

READ IN STUDENTS DATA, FROM ARCH3_TAPE.

READ (>, 1006, FNN=400) ID, DAYCD, HOUR, MIN, NDAY, NDAY*

CHECK WHETHER LAST RECORD.

IF NOT CHECK TIMES OUT OF RANGE.

220 IF (HOUR=24) 230-225-255
225 HOUR = 0
230 IF (MIN=60) 235-235-255
235 IF (MIN=60) 240-240-255
240 MIN = 0
245 IF (NDAY=60) 260-260-255
255 KINT = KINT+1
WHITE (6, 1012) ID, IACT, DAYCD, HOUR, MIN, NDAY, NDAY, KINT, KTOT
Go To 200

CHECK EACH HOURLY TIME FOR INDEXING PURPOSES.
260 HOUR = HOUR + 1
   NOHR = NOHR + 1

   CHECK OUT THE ACTIVITY CLASSIFICATIONS.

   IF STARTING AND ENDING HOURS ARE EQUAL, JUST SUBTRACT MINUTES.

   300 IF (NOHR = HOUR) 310 320 330 331
   310 T = NUMIN - MIN
       COUNT1 = T/60.0
       COUNT(IACT,DAYCD,HOUR) = COUNT(IACT,DAYCD,HOUR) + COUNT1
       LSW = 1
       GO TO 350
   320 WHEN ENDING HOUR IS LARGER, COMPUTE AND ADVANCE STARTING HOUR.
       UNTIL WE GET EQUALITY.
       320 T = 60 - MIN
           COUNT1 = T/60.0
           COUNT(IACT,DAYCD,HOUR) = COUNT(IACT,DAYCD,HOUR) + COUNT1
           LSW = 2
           GO TO 350

       ENDING HOUR IS NEXT DAY, COMPUTE AND ADVANCE STARTING HOUR.
       UNTIL WE REACH MIDNIGHT, THEN MOVE TO NEXT DAY,
       AND WORK UP THE ENDING HOUR.

   330 T = 60 - MIN
       COUNT1 = T/60.0
       COUNT(IACT,DAYCD,HOUR) = COUNT(IACT,DAYCD,HOUR) + COUNT1
       LSW = 4
       340 CONTINUE
       TOT(IACT,DAYCD) = TOT(IACT,DAYCD) + COUNT1
       ACTTOT(IACT) = ACTTOT(IACT) + COUNT1
       DTTOT(DAYCD) = DTTOT(DAYCD) + COUNT1
       GO TO 10 (200+355+360),LSW

   350 HOUR = HOUR + 1
       MIN = 0
       GO TO 300

   360 HOUR = HOUR + 1
       MIN = 0
       365 IF (HOUR = 24) 300,300,375

   370 HOUR = 1
       DAYCD = DAYCD + 1
       IF (DAYCD = 8) 300,385,385

   380 DAYCD = 1
       GO TO 300

   LET S PRINT THE STUFF WE STORED.

   A FEW SCALE VALUES WILL HELP IN READING THE GRAPH.

   WRITE (*,1008) HEAD, (ACTNM(IACT,II),II=1,2)
   DD 600 T=1,7

   WRITE (*,1009) (DAY(I,II),II=1,2),SCALE
   WRITE (*,1013)
   WRITE (*,1014)
DO 550 J=1,24
C LETS_THY_SOME_DOES_EVERY_COLUMN.
C DO 450 K=1,100
C XLINE(K) = OUTS(K)
C 450 CONTINUE
C NUM = COUNT(IACTY,I,J) + 0.5
C WE DON'T WANT TO SHOW ANYTHING WHEN NOTHING SHOULD BE THERE.
C IF (NUM.LE.0) GO TO 510
C IF (NUM.GT.100) NUM = 100
C DO 500 K=1,NUM
C XLINE(K) = X
C 500 CONTINUE
C WRITE (6,1010) XTIME(J),COUNT(IACTY,I,J),XLINE
C 550 CONTINUE
C WE CAN GET TWO DAYS TO A PAGE, SO LET S DO IT.
C IN = [1-2*(1/2)]
C IF (IN).GT.545.545*100
C WRITE (6,1011)
C WRITE (6,1000) HEAD,(ACTNAM(IACTY,I),II=1,2)
C 600 CONTINUE
C IF (IACTY = 11) A09*610*610
C WRITE (6,1010) XTIME(J),COUNT(IACTY,I,J),XLINE
C 650 CONTINUE
C IF WE ARE LUCKY, WE ARE DONE HERE.
C WRITE (6,1015)
C WRITE (6,1016)
C WRITE (6,1017)
C WRITE (6,1018)
C DO /50 IC=1,11
C DO /DU 1=1,7
C (TOT(L)) = TOT(TOT(L)) + 0.5
C PCTC(L)= (TOT(L)*TOT(L)/100.0) + 0.5
C PCTC(L) = (TOT(TOT(L))/100.0/ACTD(TOT(L))/0.5
C 700 CONTINUE
C WRITE (6,1019) (ACTNMAM(IACTY,II),II=1,7),TTO(TOTAIL),PCTC(II)
C 750 CONTINUE
C WRITE (6,1020) (DYTOL(KD),KD=1,7),GTD
C IF TLE = GTOL/168
C WRITE (6,1021) IF IF
C WRITE (6,1022) KD TLE
C FORMAT STATEMENTS.
C 1000 FORMAT (13A6)
C 1001 FORMAT (12A6/12A6)
C 1002 FORMAT (12A6)
C 1003 FORMAT (50A1)
C 1004 FORMAT (14A1)
C 1005 FORMAT (14A1)
C 1006 FORMAT (13A6)
C 1007 FORMAT (12A6)
C 1008 FORMAT (13A6/40A1)
C 1009 FORMAT (13A6/40A1)
C 1010 FORMAT (13A6/40A1)
1010 FORMAT (2X, A6, F6.2, 2X, 100A1)
1011 FORMAT (1H1)
1012 FORMAT (1X, "BAD RECORD " , 5X, 15X, 2X, 12X, 10X, 12X, 1X, 12X, 1X, 12X, 16X)
1013 FORMAT (1H1)
1014 FORMAT (1X, "HOUR")
1015 FORMAT (1H1)
1016 FORMAT (1X, "ACTIVITY")
1017 FORMAT (1X, "TOTAL")
1018 FORMAT (21X, "TOTAL")
1019 FORMAT (1X, "TOTAL")
1020 FORMAT (1X, "TOTAL HOURS")
1021 FORMAT (1X, "F.T.E. STUDENTS")
1022 FORMAT (1H1, 1X, "TOTAL NUMBER OF ACTIVITY NUMBERS GREATER THAN 26 =")
1500 FORMAT (1X, "PROCESSTIME = "F12.3,"MIN.")
<table>
<thead>
<tr>
<th>HOUR MAN-HOURS</th>
<th>10</th>
<th>20</th>
<th>30</th>
<th>40</th>
<th>50</th>
<th>60</th>
<th>70</th>
<th>80</th>
<th>90</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>24.00</td>
<td>4.83</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.00</td>
<td>5.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.00</td>
<td>3.83</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.00</td>
<td>1.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.00</td>
<td>1.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7.00</td>
<td>0.67</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8.00</td>
<td>15.40</td>
<td>XXX</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9.00</td>
<td>13.05</td>
<td>XXX</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10.00</td>
<td>3.42</td>
<td>XXX</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11.00</td>
<td>2.20</td>
<td>XXX</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12.00</td>
<td>1.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13.00</td>
<td>8.70</td>
<td>XXX</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14.00</td>
<td>8.00</td>
<td>XXX</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15.00</td>
<td>10.00</td>
<td>XXX</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16.00</td>
<td>11.00</td>
<td>XXX</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17.00</td>
<td>14.20</td>
<td>XXX</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18.00</td>
<td>2.25</td>
<td>XXX</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19.00</td>
<td>4.75</td>
<td>XXX</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20.00</td>
<td>6.75</td>
<td>XXX</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21.00</td>
<td>8.00</td>
<td>XXX</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22.00</td>
<td>7.92</td>
<td>XXX</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>23.00</td>
<td>3.92</td>
<td>XXX</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
FILE_3*ARCH3*UNIT=TAPE*BLOCKING=255*RECORD=A
FILE_5*ARCH5*UNIT=READER
FILE_6*PRINT*UNIT=PRINT
C FREQUENCY DISTRIBUTION OF ACTIVITIES, GROUPED IN 10 MINUTES
C INTERVAL, AND CENTRAL TENDENCY MEASUREMENTS.
C
C WHEN CHANGE IN NUMBER OF ACTIVITIES, CHANGE DIMENSIONS OF ARRAYS
C FOR(J), AND ACTNAM(I,J) ALSO CHANGE THE PARAMETERS OF ALL THE
C DU LOOPS RELATED.
C ALSO CHANGE CARD "NACT".
C
C WHEN CHANGE IN SIZE OF INTERVALS, CHANGE THE DIMENSIONS OF ARRAYS
C PACT(I,1), XINT(I,1), MODE(I), AND MPT(I), ALSO CHANGE CARDS
C XINT(I,2), SINT, 15INT, ALSO THE PARAMETERS OF LOOPS RELATED.
C CHANGE CARD " NINT" AND "NIV".
C
C WHEN NOT INTERVAL IS USED SET CASE! IN .TRUE.
C OTHERWISE SET IT .FALSE..
C
ST Ai R T O F SEG M E N T ********** 1

DIMENSION FQ(11,48), PACT(48), DOTS(10), CFQ(48), XLINE(100)
DIMENSION HEAD(13), SCALE(19), XINT(48,2), ACTNAM(11,2)
REAL XMOD(48), MPT(48), MEAN, LB, MEDIAN
INTEGER TMP1, TEMP, HR
LOGICAL CASE!
DATA DOTS/9, 1H /", "//"/"X"
1000 FORMAT (13A6)
1001 FORMAT (14A5)
1002 FORMAT (12A6)
1003 FORMAT (3X, 12X, 4X, 12X, 15X)
1004 FORMAT (1H1)
1005 FORMAT (1X, **BAD RECORD ******"3X, 19X, 2(12X, 12X, 12X), 10X, 15)
1006 FORMAT (1H0, 13X, 40X, 2A6)
1007 FORMAT (1H0, 28X, "0"X, 19A5)
1008 FORMAT (1H0, "INTERVAL FACT", CFQ, PERCENT")
1009 FORMAT (1H0)
1010 FORMAT (1X, F4.0, **F4.0, F6.0, 1X, F6.1X, "", 100A1)
10001 FORMAT (// // // /)
10011 FORMAT (1X, **RADE**)
1011 FORMAT (/19X, **MEAN", 17X, "MEDIAN", 15X, "VARIANCE", 13X, "STD.", "DEV"
115X, " месяцев")
1012 FORMAT (/16X, 5(F10.2, 12X))
1013 FORMAT (1H1, 103X, F10.2, /)
1014 FORMAT (1X, **MINIMUM**)
1015 FORMAT (1X, **MAXIMUM**)
1016 FORMAT (1X, **PROCESS TIME =**, F10.2, " MINUTES")
1017 FORMAT (1X, **DATA COLLECTED BY TEN MINUTES. ************")

C READ (5,1000) HEAD
C READ (5,1001) SCALE
C READ (5,1002) (ACTNAM(I,J), J=1,2, I=1,11)
C NACT = 11
NINT = 40
MIN = 6
SINT = 10, 0
ISINT = 10
XINT(I,1) = 1,0
XINT(1,2) = 10,0
CASE1 = TRUE

INITIALIZE STORAGE LOCATIONS TO ZERO, AND INITIAL VALUES NEEDED.
DO 50 I=1,NACT
DO 50 J=1,NINT
F(J) = 0.0
50 CONTINUE
DO 100 K=1,NINT
XMIN(K) = 0.0
CFD(K) = 0.0
PACT(K) = 0.0
NPT(K) = 0.0
100 CONTINUE
MEAN = 0.0
LY = 0.0
MEDIAN = 0.0
XINT = 0
NPT(1) = XINT(1,2)/2.0
DO 150 J=2,NINT
XINT(J) = XINT(J-1) + SINT
NPT(J) = NPT(J-1) + SINT
XINT(J,2) = XINT(J-1,2) + SINT
150 CONTINUE
WRITE (6,1004)
READ STUDENTS DATA FROM RICE TAPE.
200 READ (3,1003,END=400) IACT,HOUR,MIN,NDHR,NDMIN
COME HERE WHEN ACTIVITIES ARE OK, CHECK START AND ENDING TIME.
210 IF (NDHR = HOUR) 215,220,225
215 KTOH = KTOT + 1
WRITE (6,1005) IACT,HOUR,MIN,NDHR,NDMIN,KTOT
GO TO 200
WHEN STARTING AND ENDING TIME IS EQUAL, JUST SUBTRACT MINUTES.
220 TEMP1 = 0
GO TO 230
WHEN ENDING TIME IS GREATER THAN STARTING TIME, SUBTRACT
HOUR FROM NDHR, AND DET INTERVALS.
225 TEMP1 = (NDHR-HOUR)*MIN
CHECK FOR STARTING AND ENDING MINUTES.
230 IF (NDMIN = MIN) 235,240,245
WHEN ENDING MINUTES IS LESS THAN STARTING MINUTES, SUBTRACT
NDMIN FROM MIN, AND DETERMINE THE INTERVALS.
235 TEMP2 = (MIN - NDMIN)/SINT
INT = TEMP1 - TEMP2
GO TO 255
C WHEN STARTING AND ENDING MINUTES ARE EQUAL, AND NOHR AND HOUR
C ALSO, LET US KNOW IT AND SEND US A MESSAGE.

240 TEMP2 = 0
GO TO 250
C WHEN STARTING MINUTES IS LESS THAN ENDING MINUTES, SUBTRACT
C MIN FROM NDIN, AND DETERMINE THE INTERVALS.

245 TEMP2 = (NDIN - MIN)/SINT + 0.99
250 INT = TEMPI * TEMP2
C CHECK WHETHER INTERVAL IS OUT OF RANGE.

260 IF (INT < 48) INT = 48
300 IF (TIME(1)) = INT + 1.0
GO TO 200
C WHEN DATA HAS BEEN READ, COME HERE TO CONTINUE COMPUTATIONS.

400 DO 900 I=1,NACT
MEAN = 0.0
MEDIAN = 0.0
VAR = 0.0
WRITE (6,1004)
WRITE (6,1006) HEAD,(ACTNAM(I,II),II=1,2)
WRITE (6,1007) SCALE
WRITE (6,1008)
WRITE (6,1009)
CFR(1) = F0(I,1)
DO 425 J = 2,NINT
CFR(J) = CFR(J-1) + FR(I,J)
425 CONTINUE
TFQ = CFR(NINT)
DO 450 J = 1,NINT
IF (CASE) MPT(J) = XINT (J,2)
MEAN = MEAN + MPT(J)*FR(I,J)
VAR = VAR + (MPT(J)**2)*FR(I,J)
450 CONTINUE
MEAN = MEAN/TFQ
VAR = VAR/TFQ - MEAN**2
STDV = SQRT(VAR)
DU 600 K=1,NINT
PACT(K) = FQ(1,K)*100/TFQ
NUM = PACT(K) + 0.5
KK = 0
475 DD 500 JK=1,10
XLINE(KK + JK) = DOTS(JK).
500 CONTINUE
KK = KK + 10
IF (kk<LT 100) GO TO 475
IF (NUM) 560 560 510
510 DD 550, L=1, NUM
XLINE(L) = X
550 CONTINUE
.C PRINT THE STUFF STORED AND THE GRAPH.
C 560 WRITE (6,1010) (XINT(K,II),II=1,2),FR(I,K),CFR(K),PACT(K),XLINE
C 600 CONTINUE
IF (CASE1) WRITE (6,1017)
WHEN GRAPH HAS BEEN PLOTTED, COME HERE AND COMPUTE THE MEDIAN.
FIND THE INTERVAL CLASS CONTAINING THE MEDIAN.
DUM2 = TFO/2.0
IF (CASE1) GO TO 691
DO 652 K = NINT
IF (DUM2-CFO(K)) 650*660*650
650 CONTINUE
K = NINT
GO TO 660
WHEN VALUE IS EQUAL TO CFO, JUST ADD 0.5 TO UPPER LIMIT.
MEDIAN = XINT(K,2) + 0.5
GO TO 700
WHEN INTERVAL IS FOUND, COME HERE AND COMPUTE FINAL VALUE.
CHECK WHEN MEDIAN LIES IN THE FIRST INTERVAL.
MEDIAN = SINT*(DUM2/FQ(I+K))
GO TO 700
MEDIAN = LA + SINT*(DUM2-CFO(K-1))/FQ(I+K)
GO TO 700
DO 692 K = 1,NINT
IF (DUM2,LE.,CFO(K)) GO TO 693.
CONTINUE
K = NINT
GO TO 695
IFQ = 1F0
ITEST = ITFO-2*(ITFO/2)
IF (ITEST,NE,0) GO TO 695
IF (DUM2,NE,CFO(K)) GO TO 695
MEDIAN = (XINT(K+2) + XINT(K+1,2))/2.0
GO TO 700
MEDIAN = XINT(K,2)
WHEN MEDIAN IS READY, COME HERE AND COMPUTE MODE.
SEARCH FOR THE HIGHEST FREQUENCY VALUE.
N = 2
N = 1
IK = 1
FOMX = FO(I+1)
IM = 1
DO 750 J = NN, NINT
IF (FOMX,LE,FO(I+J)) 740, 750, 730
730 IM = IM + 1
GO TO 750
740 N = J
FOMX = FO(I+J)
IM = 1
750 CONTINUE
IF (CASE1) GO TO 805
<table>
<thead>
<tr>
<th>INTERVAL</th>
<th>FREQUENCY</th>
<th>CFREQUENCY</th>
<th>PERCENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0-10.</td>
<td>0.</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>11.0-20.</td>
<td>1.</td>
<td>1.00</td>
<td>10.00</td>
</tr>
<tr>
<td>21.0-30.</td>
<td>2.</td>
<td>3.00</td>
<td>30.00</td>
</tr>
<tr>
<td>31.0-40.</td>
<td>3.</td>
<td>6.00</td>
<td>60.00</td>
</tr>
<tr>
<td>41.0-50.</td>
<td>4.</td>
<td>10.00</td>
<td>100.00</td>
</tr>
<tr>
<td>51.0-60.</td>
<td>5.</td>
<td>15.00</td>
<td></td>
</tr>
<tr>
<td>61.0-70.</td>
<td>6.</td>
<td>21.00</td>
<td></td>
</tr>
<tr>
<td>71.0-80.</td>
<td>7.</td>
<td>28.00</td>
<td></td>
</tr>
<tr>
<td>81.0-90.</td>
<td>8.</td>
<td>36.00</td>
<td></td>
</tr>
<tr>
<td>91.0-100.</td>
<td>9.</td>
<td>45.00</td>
<td></td>
</tr>
</tbody>
</table>

**DATA COLLECTED BY TEN MINUTES.**********
<table>
<thead>
<tr>
<th>MEAN</th>
<th>MEDIAN</th>
<th>VARIANCE</th>
<th>STD. DEV.</th>
<th>MODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>95.33</td>
<td>60.00</td>
<td>5051.56</td>
<td>71.07</td>
<td>60.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>80.00</td>
</tr>
</tbody>
</table>
READ (5,AGUS)
READ (5,CARH)
READ (5,RRNR)
READ (5,ITN)
READ (5,TITN)
READ (5,ITEN)
READ (5,TITN)
READ (5,ITEN)
READ (5,TITN)
READ (5,TITN)
READ (5,ITEN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
READ (5,TITN)
<table>
<thead>
<tr>
<th>Time</th>
<th>Man-Hours</th>
<th>0</th>
<th>10</th>
<th>20</th>
<th>30</th>
<th>40</th>
<th>50</th>
<th>60</th>
<th>70</th>
<th>80</th>
<th>90</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>24.00</td>
<td>1.75</td>
<td>XX</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.00</td>
<td>1.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.00</td>
<td>1.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.00</td>
<td>1.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.00</td>
<td>1.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.00</td>
<td>0.50</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7.00</td>
<td>0.33</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8.00</td>
<td>4.75</td>
<td>XXXX</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9.00</td>
<td>7.92</td>
<td>XXXXXXX</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10.00</td>
<td>6.42</td>
<td>XXXXX</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11.00</td>
<td>1.55</td>
<td>XX</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12.00</td>
<td>1.75</td>
<td>XX</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13.00</td>
<td>9.42</td>
<td>XXXXXXX</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14.00</td>
<td>11.93</td>
<td>XXXXXXX</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15.00</td>
<td>16.50</td>
<td>XXXXXXXXXX</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16.00</td>
<td>15.08</td>
<td>XXXXXXXXXX</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17.00</td>
<td>3.55</td>
<td>XXXXXXXX</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18.00</td>
<td>1.42</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19.00</td>
<td>4.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20.00</td>
<td>6.43</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21.00</td>
<td>2.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22.00</td>
<td>6.02</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>23.00</td>
<td>3.12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Thursday**

<table>
<thead>
<tr>
<th>Time</th>
<th>Man-Hours</th>
<th>0</th>
<th>10</th>
<th>20</th>
<th>30</th>
<th>40</th>
<th>50</th>
<th>60</th>
<th>70</th>
<th>80</th>
<th>90</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>24.00</td>
<td>4.50</td>
<td>XXXX</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.00</td>
<td>4.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.00</td>
<td>1.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.00</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7.00</td>
<td>0.43</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8.00</td>
<td>15.75</td>
<td>XXXXXXXXXX</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9.00</td>
<td>8.45</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10.00</td>
<td>6.22</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11.00</td>
<td>6.11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12.00</td>
<td>2.75</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13.00</td>
<td>28.34</td>
<td>XXXXXXXXXXXXXXXXXXXX</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14.00</td>
<td>28.47</td>
<td>XXXXXXXXXXXXXXXXXXXX</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15.00</td>
<td>25.43</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16.00</td>
<td>23.83</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17.00</td>
<td>20.17</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18.00</td>
<td>1.95</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19.00</td>
<td>4.75</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20.00</td>
<td>1.75</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21.00</td>
<td>6.25</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22.00</td>
<td>5.92</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>23.00</td>
<td>3.14</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
FILE SEARCH UNIT=TACE, BLOCKING=255, RECORD=4
FILE SEARCH UNIT=TOACHE
FILE ASKJPY UNIT=PRINT
C
C PROGRAM NAME: ENCODER/ARCH
C INPUT= ARCH3 TAPE
C OUTPUT= PRINTER TAPE.
C
C WRITTEN FOR THE PURPOSE OF COLLECTING AND ANALYSING
C
C SUMMARY ON MAN-HOURS LOAD FOR EACH SPACE AND ACTIVITY.
C
C DECLARATIONS
C
C DIMENSION CM(26,11,7), TOTSA(26,7), TOTA(11,7), TOTSAC(26,117)
C
C START OF SEGMENT
C
C DATA CM: TOTSA, TOTA(26,0,1,2,0,7,0,0,7,0,0,20,0,0)/
C DATA TOTSAC: TOTA(26,0,1,2,0,7,0,0,7,0,0,20,0,0)/
C DATA C:
C
C FORNATS
C
C 1000 FORMAT (1X6)
C 1005 FORMAT (14X6)
C 1010 FORMAT (14X6)
C 1015 FORMAT (14X6)
C 1020 FORMAT (14X6)
C 1025 FORMAT (14X6)
C 1030 FORMAT (14X6)
C 1035 FORMAT (14X6)
C 1040 FORMAT (14X6)
C 1045 FORMAT (14X6)
C 1050 FORMAT (14X6)
C 2000 FORMAT (1X11)
C 2005 FORMAT (1X11)
C 2010 FORMAT (1X11)
C 2015 FORMAT (1X11)
C 2020 FORMAT (1X11)
C 2025 FORMAT (1X11)
C 2030 FORMAT (1X11)
C 2035 FORMAT (1X11)
C 2040 FORMAT (1X11)
C 2045 FORMAT (1X11)
C 2050 FORMAT (1X11)
C 2055 FORMAT (1X11)
C 2060 FORMAT (1X11)
C 3000 FORMAT (1X11)
C
C SEGMENT 215 125 LONG
C
C CHECK YOUR TAPE. SAVE IT BEFORE REWIND. CAREFUL.
C
C END OF PROGRAM.
C

C \textbf{HEAD (5\times1000) HEAD}
\textbf{READ (5\times1000) ((DYNOM(T,J), J=1,2), T=1,7)}
\textbf{READ (5\times1000) ((ACTAM(T,J), J=1,2), T=1,ACT)}
\textbf{READ (5\times1000) ((SPCNAM(T,J), J=1,2), T=1,NSPAC)}
\textbf{WRITE (6\times1000))

C

C \textbf{HEAD IN SIMPNTS DATA, INPUT TAPE.}
\textbf{READ (3\times1024, E00=5000) ID\times ACT\times DAY\times HU\times MIN\times DU\times N\times N\times N\times ISPAC}

C

C \textbf{IF (HU\times LT, IS>AU\times LT\times HU\times ) GO TO 125}
\textbf{IF (IS>MIN, IS=MIN)

C

C \textbf{105 T2 = (MIN - MIN)/60,}
\textbf{T3 = T1 - T2}
\textbf{GU 11 300}

C

C \textbf{110 T2 = 0,

C \textbf{120 T3 = T1 + T2}
\textbf{GU 11 300}

C

C \textbf{125 K = K + 1}
\textbf{WHILE (6\times2000) ID\times ACT\times DAY\times HU\times MIN\times DU\times N\times N\times N\times ISPAC\times K}
\textbf{GU 11 200}

C

C \textbf{300 CONTINUE}

C

C \textbf{CONTINUE HERE TO FILL MATRIX OF TIMES.}
\textbf{CNT(ISPAC\times ACT\times DAY) = CNT(ISPAC\times ACT\times DAY) + T3}
\textbf{TU\times IS\times (ISPAC\times DAY) = T\times ID\times IS\times (ISPAC\times DAY) + T3}
\textbf{TU\times T\times S(ISPAC\times ACT) = T\times U\times S(ISPAC\times ACT) + T3}
\textbf{T\times T\times U\times S(ISPAC) = T\times T\times U\times S(ISPAC) + T3}
\textbf{T\times T\times U\times S(DAY) = T\times T\times U\times S(DAY) + T3}
\textbf{HIGH\times I = HIGH\times I + T3}
\textbf{GU 11 200}

C

C \textbf{END DATA HAS BEEN READ, NOW PRINT TABLES IN US.}
\textbf{5000 GU 370 IA = 1\times ACT}
\textbf{WRITE (6\times1050)

C

C \textbf{WRITE (6\times2005)

C

C \textbf{WRITE (6\times2015)

C

C \textbf{WRITE (6\times2020)

C

C \textbf{DO 360 IS = 1,ISPAC}

C

C \textbf{TUT = 0,0}
\textbf{DO 350 IDY = 1,7}
\textbf{TUT = TUT + CNT(IS,IA,IDY)}

C

C \textbf{350 CONTINUE}
\textbf{WRITE (6\times2025) (SPC\times (IS,II), II = 1,2), (CNT(IS,IA,IDY), IDY = 1,7), TUT)

C

C \textbf{360 CONTINUE}
\textbf{WRITE (6\times2020) (TOT\times (TA,IDY), IDY = 1,7), (T\times I\times A (IA))}
C     COME HERE AFTER TABLES, AND PRINT A SUMMARY.
C
C     WRITE (6,1050)
WRITE (6,1005)
WRITE (6,2010)
WRITE (6,2035)
WRITE (6,2020)
DO 300 IS = 1,NSPAC
WRITE (6,2025) (SPCNAM(IS,II),II=1,2),(TOTSA(IS,1D),1D=1,7),
     
1 TOTS(IS)
C     CONTINUE
WRITE (6,2030) (TOTS(1D),1D=1,7),BIGTOT
C     COME HERE WHEN TABLES TYPE ONE ARE READY, AND CONTINUE
C     COMPUTATIONS FOR TABLES TYPE TWO.
C
C     WRITE (6,1050)
DO 420 INY = 1,7
WRITE (6,1050)
WRITE (6,1045)
WRITE (6,1040)
WRITE (6,2040)
WRITE (6,2045)
DO 410 IS = 1,NSPAC
INT = 0,0
DO 400 IA = 1,NAC1
TOL = TOL + CNCT(IS,IA,1D)
400 CONTINUE
WRITE (6,2050) (SPCNAM(IS,II),II=1,2),(CNCT(IS,IA,1D),IA=1,NACT),
     
1 TOL TOL
410 CONTINUE
WRITE (6,2055) (TOTSA(IS,1D),1D=1,NACT),INTDC(1D)
C     CONTINUE
WRITE (6,2055)
C     COME HERE AND PRINT SUMMARY OF TABLES TYPE TWO.
C
C     WRITE (6,1050)
WRITE (6,1005)
WRITE (6,2010)
WRITE (6,2060)
WRITE (6,2040)
DO 300 IS = 1,NSPAC
WRITE (6,2050) (SPCNAM(IS,II),II=1,2),(TOTSA(IS,IA),IA=1,NACT),
     
1 TOTS(IS)
C     CONTINUE
WRITE (6,2055) (TOTSA(IA),IA=1,NACT),BIGTOT
C     OK, IF I HAVE BEEN SMART, I AM THERE, OTHERWISE I AM FINISHED.
C
C     PTIME = TIME (PTIME)
WRITE (6,3000) PTIME
STOP
END

SEGMENT 1 IS 720 LONG.
### TABLES OF MAN-HOUR LOADS FOR EACH SPACE, BY ACTIVITY AND DAYS

#### MAN-HOURS LOAD FOR EACH SPACE VS ACTIVITY PER DAY

<table>
<thead>
<tr>
<th>SPACE</th>
<th>LECTURE</th>
<th>PLAY</th>
<th>STUDIO</th>
<th>STUDY</th>
<th>SLEEP</th>
<th>EAT</th>
<th>PHYSICAL</th>
<th>REC</th>
<th>W.K. BYKE</th>
<th>BUS</th>
<th>CAN</th>
<th>MISC</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weiss Coll</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>14.67</td>
<td>11.58</td>
<td>8.58</td>
<td>11.00</td>
<td>14.08</td>
<td>0.00</td>
<td>0.50</td>
<td>6.07</td>
<td>0.00</td>
<td>64.08</td>
</tr>
<tr>
<td>Hansen Coll</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>9.50</td>
<td>0.00</td>
<td>0.83</td>
<td>1.00</td>
<td>3.67</td>
<td>0.00</td>
<td>0.00</td>
<td>2.67</td>
<td>0.00</td>
<td>17.67</td>
</tr>
<tr>
<td>Will Coll</td>
<td>0.00</td>
<td>0.00</td>
<td>1.50</td>
<td>0.25</td>
<td>5.75</td>
<td>1.17</td>
<td>0.42</td>
<td>1.50</td>
<td>0.00</td>
<td>0.00</td>
<td>1.05</td>
<td>0.00</td>
<td>12.17</td>
</tr>
<tr>
<td>Baker Coll</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>7.00</td>
<td>22.33</td>
<td>2.17</td>
<td>1.08</td>
<td>6.25</td>
<td>0.00</td>
<td>0.00</td>
<td>4.67</td>
<td>0.00</td>
<td>10.54</td>
</tr>
<tr>
<td>Jones Coll</td>
<td>1.33</td>
<td>0.00</td>
<td>3.25</td>
<td>1.00</td>
<td>6.75</td>
<td>1.82</td>
<td>0.25</td>
<td>5.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.75</td>
<td>0.00</td>
<td>10.95</td>
</tr>
<tr>
<td>Love Coll</td>
<td>1.00</td>
<td>0.00</td>
<td>1.00</td>
<td>6.17</td>
<td>2.17</td>
<td>1.20</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>1.00</td>
<td>0.00</td>
<td>7.33</td>
</tr>
<tr>
<td>Love Hall</td>
<td>2.17</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>2.17</td>
</tr>
<tr>
<td>Anderson H</td>
<td>20.42</td>
<td>0.75</td>
<td>82.33</td>
<td>14.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.17</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>4.00</td>
<td>0.00</td>
<td>123.67</td>
</tr>
<tr>
<td>Nayfair Hall</td>
<td>3.92</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.50</td>
<td>0.00</td>
<td>4.42</td>
</tr>
<tr>
<td>Mamman Hall</td>
<td>2.83</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>2.83</td>
</tr>
<tr>
<td>Chemistry L</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Mechanical L</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Physics Lab</td>
<td>5.75</td>
<td>0.00</td>
<td>0.00</td>
<td>0.25</td>
<td>0.00</td>
<td>0.25</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>6.75</td>
</tr>
<tr>
<td>Allen Cent</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Herman Halls</td>
<td>3.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>3.00</td>
</tr>
<tr>
<td>Anti Building</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Ryun Lab</td>
<td>4.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>4.00</td>
</tr>
<tr>
<td>Memorial Cnt</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>3.42</td>
<td>0.83</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>1.17</td>
<td>0.00</td>
<td>4.42</td>
</tr>
<tr>
<td>Gymnasium H</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>2.00</td>
<td>0.00</td>
<td>2.00</td>
<td>0.00</td>
<td>0.00</td>
<td>2.00</td>
<td>0.00</td>
<td>2.00</td>
</tr>
<tr>
<td>Stadium U</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Undef SD</td>
<td>8.04</td>
<td>1.00</td>
<td>9.67</td>
<td>6.25</td>
<td>36.75</td>
<td>4.42</td>
<td>4.33</td>
<td>4.42</td>
<td>0.00</td>
<td>0.00</td>
<td>14.47</td>
<td>0.00</td>
<td>93.17</td>
</tr>
<tr>
<td>Move on Camp</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.42</td>
<td>17.17</td>
<td>2.58</td>
<td>22.17</td>
<td>57.17</td>
<td>27.17</td>
<td>21.57</td>
<td>93.17</td>
</tr>
<tr>
<td>Off Campus</td>
<td>0.00</td>
<td>0.00</td>
<td>4.83</td>
<td>29.56</td>
<td>105.12</td>
<td>18.20</td>
<td>8.85</td>
<td>27.50</td>
<td>2.58</td>
<td>22.17</td>
<td>57.17</td>
<td>27.17</td>
<td>271.00</td>
</tr>
<tr>
<td>Old Stadium</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>55.04</td>
<td>2.25</td>
<td>122.28</td>
<td>100.33</td>
<td>207.37</td>
<td>46.40</td>
<td>28.85</td>
<td>75.87</td>
<td>20.25</td>
<td>25.77</td>
<td>111.22</td>
<td>795.67</td>
<td>795.67</td>
</tr>
</tbody>
</table>
TABLES OF MAN-HOUR LOADS FOR EACH SPACE, BY ACTIVITY AND DAYS. - RICE PROJECT

MAN-HOURS LOAD FOR EACH SPACE VS DAYS PER ACTIVITY.

<table>
<thead>
<tr>
<th>SPACE</th>
<th>MONDAY</th>
<th>TUESDAY</th>
<th>WEDNESDAY</th>
<th>THURSDAY</th>
<th>FRIDAY</th>
<th>SATURDAY</th>
<th>SUNDAY</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weiss Coll</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>4.17</td>
<td>0.00</td>
<td>0.00</td>
<td>4.17</td>
</tr>
<tr>
<td>Hahn Coll</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Mill Coll</td>
<td>0.00</td>
<td>2.08</td>
<td>1.50</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>9.00</td>
<td>2.25</td>
</tr>
<tr>
<td>Baker Coll</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Jones Coll</td>
<td>0.00</td>
<td>3.17</td>
<td>3.25</td>
<td>7.00</td>
<td>7.00</td>
<td>0.00</td>
<td>0.00</td>
<td>20.42</td>
</tr>
<tr>
<td>Brown Coll</td>
<td>4.33</td>
<td>3.67</td>
<td>0.00</td>
<td>0.03</td>
<td>0.00</td>
<td>0.00</td>
<td>10.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Lovett Coll</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>4.58</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>4.58</td>
</tr>
<tr>
<td>Lovett Hall</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Anderson Hall</td>
<td>71.67</td>
<td>56.75</td>
<td>64.33</td>
<td>141.97</td>
<td>75.22</td>
<td>14.38</td>
<td>31.25</td>
<td>515.57</td>
</tr>
<tr>
<td>Kaye Hall</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Hannan Hall</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Forefen Lab</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>4.75</td>
<td>0.00</td>
<td>0.00</td>
<td>4.75</td>
</tr>
<tr>
<td>Chemistry Lab</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Mechanical Lab</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Physics Lab</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Allen Center</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>3.33</td>
<td>0.00</td>
<td>3.33</td>
</tr>
<tr>
<td>Herman Brown</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Art Building</td>
<td>23.88</td>
<td>1.33</td>
<td>13.70</td>
<td>4.17</td>
<td>26.42</td>
<td>0.00</td>
<td>1.25</td>
<td>73.45</td>
</tr>
<tr>
<td>Rydlin Hall</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Memorial Gym</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>2.06</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>2.06</td>
</tr>
<tr>
<td>Gymnasium H</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Stadium H</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Undefined SP</td>
<td>11.67</td>
<td>10.54</td>
<td>9.67</td>
<td>9.42</td>
<td>13.00</td>
<td>7.92</td>
<td>3.50</td>
<td>65.28</td>
</tr>
<tr>
<td>Move on Camp</td>
<td>4.08</td>
<td>4.92</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>7.00</td>
<td>4.08</td>
</tr>
<tr>
<td>Diff Campus</td>
<td>4.17</td>
<td>4.84</td>
<td>6.83</td>
<td>0.00</td>
<td>0.00</td>
<td>2.25</td>
<td>8.23</td>
<td>24.07</td>
</tr>
<tr>
<td>Old Stadium</td>
<td>0.00</td>
<td>7.75</td>
<td>0.00</td>
<td>5.58</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>13.33</td>
</tr>
<tr>
<td>TOTAL</td>
<td>121.79</td>
<td>140.83</td>
<td>129.20</td>
<td>177.47</td>
<td>129.22</td>
<td>28.55</td>
<td>48.48</td>
<td>758.53</td>
</tr>
</tbody>
</table>
PROGRAM NAME: HEAM1/JOIST1.

INPUT FROM CARDS Punched, CONTAINING JOISTS AND BEAMS DATA.

OUTPUT WILL CONSIST OF TWO TABLES AND A GRAPH.

THE PROGRAM IS USED TO SELECT AMONG DIFFERENT JOISTS AND BEAMS AVAILABLE, THE OPTIMUM SECTIONS FOR EACH BAY SIZE, AS WELL AS THE COSTS ASSOCIATED WITH EACH SOLUTION.

THE PROGRAM DEALS WITH TWO SYSTEMS.

A1: BEAMS SPAN LENGTH AND JOISTS SPAN WIDTH.

A2: BEAMS SPAN WIDTH AND JOISTS SPAN LENGTH.

TWO ALTERNATIVES ARE CONSIDERED.

1A FLOORS ONLY.

1B Structural elements for roofs only.

2A Structural elements for both floors.

A COMPLETE LIST OF VARIABLES, AND THEIR MEANING WILL BE PROVIDED.

IN THE DOCUMENTS ATTACHED.

GOOD LUCK.

DECLARATIONS.

START OF SEGMENT

DIMENSION ALL(33,2), ALLU(56), ACTHTJ(33), SHAPI(33), ACTTWB(56)
DIMENSION SEC(56), T(56), SHAPR(56), FALLS(2,56)
REAL X(MAX(33), MAX(56), INJ(33)), INY(33), LINF(120)
REAL LLAD, LLDA, LENGTH, QMMFX, INDEX
INTEGER V, TCI(120), TC2(120)
LOGICAL K, DEF, AUX, CARGA
LOGICAL A, R

DATA HLEN, DDT, STAR, ITAL, ZERH/14, "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", "", ""
<table>
<thead>
<tr>
<th>Line</th>
<th>Code</th>
<th>Description</th>
<th>Address</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>WRITE (66,010)</td>
<td>DISTRLOAD</td>
<td>00011720</td>
<td>0238</td>
</tr>
<tr>
<td>2</td>
<td>WRITE (66,000)</td>
<td>DISTRLOAD</td>
<td>00011720</td>
<td>0238</td>
</tr>
<tr>
<td>3</td>
<td>WRITE (66,065)</td>
<td>DISTRLOAD</td>
<td>00011720</td>
<td>0238</td>
</tr>
<tr>
<td>4</td>
<td>WRITE (66,070)</td>
<td>DISTRLOAD</td>
<td>00011720</td>
<td>0238</td>
</tr>
<tr>
<td>5</td>
<td>WRITE (66,075)</td>
<td>DISTRLOAD</td>
<td>00011720</td>
<td>0238</td>
</tr>
<tr>
<td>6</td>
<td>WRITE (66,080)</td>
<td>DISTRLOAD</td>
<td>00011720</td>
<td>0238</td>
</tr>
<tr>
<td>7</td>
<td>WRITE (66,090)</td>
<td>DISTRLOAD</td>
<td>00011720</td>
<td>0238</td>
</tr>
<tr>
<td>8</td>
<td>WRITE (66,095)</td>
<td>DISTRLOAD</td>
<td>00011720</td>
<td>0238</td>
</tr>
<tr>
<td>9</td>
<td>ALLU(1),X(1),</td>
<td>DISTRLOAD</td>
<td>00011720</td>
<td>0238</td>
</tr>
<tr>
<td>10</td>
<td>WRITE (66,065)</td>
<td>DISTRLOAD</td>
<td>00011720</td>
<td>0238</td>
</tr>
<tr>
<td>11</td>
<td>WRITE (66,070)</td>
<td>DISTRLOAD</td>
<td>00011720</td>
<td>0238</td>
</tr>
<tr>
<td>12</td>
<td>WRITE (66,075)</td>
<td>DISTRLOAD</td>
<td>00011720</td>
<td>0238</td>
</tr>
<tr>
<td>13</td>
<td>WRITE (66,080)</td>
<td>DISTRLOAD</td>
<td>00011720</td>
<td>0238</td>
</tr>
<tr>
<td>14</td>
<td>WRITE (66,090)</td>
<td>DISTRLOAD</td>
<td>00011720</td>
<td>0238</td>
</tr>
<tr>
<td>15</td>
<td>WRITE (66,095)</td>
<td>DISTRLOAD</td>
<td>00011720</td>
<td>0238</td>
</tr>
</tbody>
</table>

C COMPUTE VARIABLES THAT WILL BE USED SAME ALL PROGRAM.

XLOAD = LOAD + ULOAD
Y = 120 + 3

C START ITERATIONS FOR SPANNING METHOD.

DO 100 1 = 1,2

C NUMBER OF BAY SIZES COUNTER.

KK = 0
A = .FALSE.
IF (1 .GT. 1) GO TO 100

C COME HERE WHEN BEAMS SPAN WIDTH AND JOISTS SPAN LENGTH.

100 WRITE (66,010) | DISTRLOAD | 00011720 | 0238 |

C COME HERE WHEN BEAMS SPAN LENGTH AND JOISTS SPAN WIDTH.

100 WRITE (66,010) | DISTRLOAD | 00011720 | 0238 |

C START ITERATIONS FOR WIDTH VARIATION.

110 ON 10 | A10 W = L1*2* | DISTRLOAD | 00011720 | 0238 |

K = .TRUE.
H = .FALSE.

C START ITERATIONS FOR LENGTH VARIATION.

120 ON 10 | A10 L = H*2* | DISTRLOAD | 00011720 | 0238 |

IF (1 .LT. 1) GO TO 120
LENGTH = L
WIDTH = W
GO TO 130

C COME HERE WHEN INTERCHANGE OF LENGTH AND WIDTH.

120 LENGTH = W
WIDTH = L

C WHEN HEAJS SPAN LENGTH JOISTS ARE EQUAL FOR EACH INCREMENT OF LENGTH

C COMPUTE JUST ONE JOINT AND SKIP UNNECESSARY TIME.
130 IF (NUTX AND J.EQ.1) GO TO 220
  IF (AUNX) GO TO 305
  K = 'FALSE'
C START JUPT COMPUTATIONS.
  ESTMT = ESTJ
  LOAD = XLOAD*DIST
  ILOAD = LOAD + ESTMT
  SHLAM = (ILOAD*LENGTH)/2.0
  WMX = ((ILOAD*LENGTH+2)/8.0)*0.012
  IF (K = 'FALSE') DEFJ = (LENGTH+12.0)/FACTOR
  IF (SHLAM > GTJMAXNJ) GO TO 292
C SEARCH FOR JOISTS WHICH ALLOWABLE SPAN PERMITS ITS USE IN CURRENT CASE
C WHEN SPAN GIVES THE DECISION, KEEP TRACK.
  GO TO 150
  IF (ALLJ = ICASE,GE.LENGTH) GO TO 160
  CONTINUE
  BTST = S
  GO TO 204
  BTST = S
  DUMW = ACUTWX(NJ)
  BTST = 0
  IF (HUMFX GT JMAXNJ) OR (SHLAM GT JMAXNJ) GO TO 292
C START IERATIONS FOR JOIST SELECTION-END DETERMINE THE DECISION.
C VARIABLE WHICH ULTIMATELY GOVERNS THE CHOICE.
C AND START THE SEARCH FOR THE BEST SECTION.
  IF (CARHA) GO TO 172
  ALNAM = LOAD*DIST
  GO TO 174
  D0 190 J = N
  IF (0.0 = ALNAM) GO TO 175
  IF (ACTWX(J),GT,0.0) GO TO 190
  IF (HUMFX(J),GT,0.0) GO TO 176
  BTST = S
  GO TO 190
  IF (SHLAM(J)) 102,102,100
  BTST = S
  GO TO 190
  BTST = S
  GO TO 190
  D0 187 DELTX = (5.0*ALNAM*LENGTH**4*Y)/(384.0*EVAL*INJ(J))
  IF (D.0 = DEF) GO TO 168
  IF (DELTX GT DELTX(J)) 188,188,186
  BTST = D
  AUX = 'TRUE'
  GO TO 190
  DUMW = ACUTWX(J)
  DUMJ = SAPP(X(J))
  AUX = 'FALSE'
  BTST = 0
  CONTINUE
190 IF NO SELECTION IS FOUND USEFUL FOR DEFLECTION SEND A MESSAGE.
  IF (CAUX) GO TO 29A
C COML HERE AFTER A SEARCH IS COMPLETE AND CHECK WEIGHT.
  ARM = ESTWX = DUMWT
  IF (ANSCAUS) LE ESTWT*0.20) GO TO 210
C COML HERE WHEN ESTIMATED AND ACTUAL WEIGHT DO NOT MATCH.
  ESTWX = DUMWT
  ILOAD = LOAD + ESTMT
  SHLAM = (ILOAD*LENGTH)/2.0
  WMX = ((ILOAD*LENGTH+2)/8.0)*0.012
  BTST = 1
  GO TO 170
C COME HERE IF NEW ESTIMATED WEIGHT AND ACTUAL WEIGHT ARE EQUAL.
210  SHJ = UUMJ
     WJ = UUMNT
     DUMMY = WJ*LENGTH
C COME HERE TO START BEAM COMPUTATION.
220  IF (A*UUMJ) GO TO 305
     NMNJ = WIDTH/DIST
     ESTWT = LSTH
     AREA = LENGTH*WIDTH
     LOAD = ((NMNJ-1)*DUMMY) + AREA*LOAD
     TLOAD = LMAID + ESTWT*WIDTH
     SHEAR = FLADOV2+V
     MNYX = ((LOAD+WIDTH)/A.0)*0.012
     SMMJ = MU*FX/FALLH(1)
C CHECK IF DEFLECTION IS USED AS DECISION VARIABLE AND IF SO GET MAX VALUE.
230  IF (DLT) *DLIHN = (WIDTH+12.0)/FACTOR
     C
240  GO TO 250
     IF (ALH(+N.0)) WIDTH) GO TO 250
250  CONTINUE
     245  ATESJ = SP
     GO TO 250
260  IF (SMHD) AT.SEC(NEW)) GO TO 306
     C
270  IF (CARGA) GO TO 262
     LOAD = AREA*LLOA
     GO TO 264
262  LOAD = TLOAD
     264  GO TO 250
     IF (DIST) *S(J) GO TO 250
     IF (DIST) *S(J) GO TO 255
     IF (ACTH(J)) GETUHM(J) GO TO 250
     IF (SMHD) *SEC(J) 260+266
266  ATESJ = MN
     GO TO 260
268  ALLSHJ = FALLV+O(J)+T(J)
     IF (SHEAR=ALLSHJ) 272+272+270
     270  ATESJ = SH
     GO TO 275
272  DELTA = (S.0+ALOAD*WIDTH+3*Y)/(384.0+EVAL+INB(J))
     IF (DELTA=DELTA) GO TO 276
     IF (DELTA=DELTA) GO TO 274
     274  ATESJ = DE
     275  AUX = TRUE
     GO TO 260
276  DUMMY = SHAPE(J)
     DUMNJ = ACTH(J)
     IN = MJ(J)
     AUX = FALSE
     DTESJ = 0
     280  CONTINUE
C
285  IF (AUX) GO TO 302
C
290  AGHJS = ESTWT = DUMMT
     IF (ABS(AUCHS)+ESTWT*0.20) GO TO 280
     285  ESTWT = DUMMT
<table>
<thead>
<tr>
<th>LOAD</th>
<th>LOAD + ESTMT*WIDTH</th>
<th>00030000 T 0741</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHEAR</td>
<td>LOAD/2.0</td>
<td>00030100 T 0743</td>
</tr>
<tr>
<td>NUMFX</td>
<td>(LOAD*WIDTH)/8.0 + 0.012</td>
<td>00030200 T 0746</td>
</tr>
<tr>
<td>SHUJ</td>
<td>NUMFX/FALLUCIN</td>
<td></td>
</tr>
<tr>
<td></td>
<td>GO TO 260</td>
<td>00030300 T 0752</td>
</tr>
<tr>
<td></td>
<td></td>
<td>00030400 T 0754</td>
</tr>
<tr>
<td></td>
<td></td>
<td>00030500 T 0755</td>
</tr>
<tr>
<td></td>
<td></td>
<td>00030600 T 0756</td>
</tr>
<tr>
<td></td>
<td></td>
<td>00030700 T 0756</td>
</tr>
<tr>
<td></td>
<td></td>
<td>00030800 T 0756</td>
</tr>
<tr>
<td></td>
<td>GLH  = GLH + 1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>GO TO 310</td>
<td></td>
</tr>
</tbody>
</table>

### C1 (to determine weights and costs for joists and beams)

<table>
<thead>
<tr>
<th>TOTJ</th>
<th>(MTJ+NUMJ)/WIDTH</th>
<th>00031000 T 0757</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOTM</td>
<td>WIN/LENGTH</td>
<td>00031100 T 0759</td>
</tr>
<tr>
<td>COSTJ</td>
<td>COSTJ*PRICEJ</td>
<td></td>
</tr>
<tr>
<td>COSTH</td>
<td>COSTH*PRICEH</td>
<td></td>
</tr>
<tr>
<td>TCOSTJ</td>
<td>= COSTJ + 0.5</td>
<td>00031300 T 0761</td>
</tr>
<tr>
<td>TCOSTH</td>
<td>= COSTH + 0.5</td>
<td></td>
</tr>
<tr>
<td>ITCOSTJ</td>
<td>= TCOSTJ + 0.5</td>
<td>00031400 T 0763</td>
</tr>
<tr>
<td>ITCOSTH</td>
<td>= TCOSTH + 0.5</td>
<td></td>
</tr>
<tr>
<td>COSTJ</td>
<td>= (COSTJ/100.0)</td>
<td>00031500 T 0766</td>
</tr>
<tr>
<td>COSTH</td>
<td>= (COSTH/100.0)</td>
<td></td>
</tr>
<tr>
<td>TCOSTJ</td>
<td>= (TCOSTJ/100.0)</td>
<td></td>
</tr>
<tr>
<td>TCOSTH</td>
<td>= (TCOSTH/100.0)</td>
<td></td>
</tr>
</tbody>
</table>

### C2 (when no section was found to satisfy actual conditions)

| 292    | BTEST = S                                   | 00032000 T 0785 |
| 294    | ATEST = F                                   |                 |
|        | IF (1=LU+2) GO TO 304                       | 00032500 T 0788 |
|        | A = +TRUE                                   |                 |
| 300    | ATEST = MN                                  | 00032900 T 0791 |
| 302    | BTEST = F                                   |                 |
| 304    | B = +TRUE                                   |                 |
| 305    | SHUJ = NO                                   |                 |
|        | SHUJ = NO                                   |                 |
|        | TOTJ = 0.0                                  |                 |
|        | TOTM = 0.0                                  |                 |
|        | DELTJ = 0.0                                 |                 |
|        | DELTH = 0.0                                 |                 |
|        | DELTXJ = 0.0                                |                 |
|        | DELTMY = 0.0                                |                 |
|        | TOTJU = 0.0                                 |                 |
|        | TOTJU = 0.0                                 |                 |
|        | COSTJ = 0.0                                 |                 |
|        | COSTH = 0.0                                 |                 |
|        | ITCOSTJ = 0.0                               |                 |
|        | ITCOSTH = 0.0                               |                 |
|        | WRITE (64,4000)                             |                 |
| 310    | IF (1,LU+1) GO TO 320                       | 00033500 T 0806 |
|        | KK = KK + 1                                 |                 |
|        | TC2(KK) = ITCOST                           |                 |
| 320    | KK = KK + 1                                 |                 |
|        | TC1(KK) = ITCOST                           | 00034000 T 0815 |

### C3 (if def)

| 330    | IF (DEL) GO TO 331                         | 00034500 T 0816 |
|        | WRITE (64,4055)                            |                 |
|        | SHUJ = WTJ + DELTJ + TOTJ + COSTJ + ATEST + SHUJ + WTJ + DELTJ + TOTJ + COSTJ + ATEST + SHUJ | 00035000 T 0817 |
|        | ITOJ, COSTJ, ATEST, TCOST                 |                 |
|        | GO TO 400                                  |                 |

### C4

| 331    | WRITE (64,4050)                            | 00035500 T 0818 |
|        | SHUJ = WTJ + DELTJ + DELTXJ + TOTJ + COSTJ + ATEST + SHUJ + WTJ + DELTJ + DELTXJ + TOTJ + COSTJ + ATEST + SHUJ | 00035600 T 0819 |
|        | ITOJ, COSTJ, ATEST, TCOST                 | 00035700 T 0820 |
|        | GO TO 400                                  |                 |
400 CONTINUE
410 CONTINUE
420 CONTINUE
DO 405 N = 1, KK
430 IF (TC1(N)) 475, 470, 475
440 IF (TC2(N)) 480, 485, 490
450 MIN = TC1(N)
GO TO 490
460 MIN = TC2(N)
GO TO 490
470 MAX = 0
480 IF (MAX = TC1(L)) 500, 510, 510
500 MAX = TC1(L)
GO TO 520
510 IF (MIN .GT. TC1(L) .AND. TC1(L) .NE. 0) MIN = TC1(L)
520 IF (MAX = TC2(L)) 530, 540, 540
530 MAX = TC2(L)
GO TO 550
540 IF (MIN .GT. TC2(L) .AND. TC2(L) .NE. 0) MIN = TC2(L)
550 CONTINUE
C DETERMINE VERTICAL AXIS RANGE OF VALUES.
560 MXYXMT = YENT
INDEX = 1.0
GO TO 590
570 MXLX = VENT / MXYXMT
DO 580 L = 1, KK
TC1(L) = (TC1(L) / INDEX) + 0.5
TC2(L) = (TC2(L) / INDEX) + 0.5
580 CONTINUE
C COME HERE AND PRINT THE GRAPH OF RESULTS.
590 CALL CATHENA (TC1, TC2, MXYXMT, KK, LINE, DOT, BLANK, STAR, ITAL, ZER, LI,
IL?, 0.51, JJ, MIN, MAX, INDEX)
PRINT = TIME + TIME(2)/3600.0
WHITE (6) PTIME
STOP
END

SEGMENT 1 IS 1023 LONG
START OF SEGMENT ********** 5
SEGMENT 5 IS 46 LONG
140 LINE(JK) = ZER
150 CONTINUE
   IMAX = (MAX*INDEX) * 0.5
   SCALE = IMAX/100.0
   WHITE (6,2060) SCALE* LINE
   MAX = MAX - 1
500 CONTINUE
C COME HERE AND PRINT THE GRAPH FOOT.
   WRITE (6,2070)
   WRITE (6,2080)
   WRITE (6,2081)
   WRITE (6,2082)
   WRITE (6,2085)
   WRITE (6,2090)
C
RETURN
END

SEGMENT 6 IS 156 LONG
DIST BETWEEN JOISTS = 2.0 FT  LIVE LOAD = 50.00 LBS/SQ.FT  DEAD LOAD = 30.00 LBS/SQ.FT  EST WEIGHT JOIST = 8.0 LBS./Ft
EST WEIGHT HEM = 50.0 LBS./FT  COST OF JOISTS = 0.14 $/LB  COST OF BEAMS = 0.20 $/LB  DEFLECTION FACTOR = 360.0
L = 29000.0 KIPS/SQ.IN  FB (1) = 24.0 KIPS/SQ.IN  FB (2) = 22.0 KIPS/SQ.IN  FV = 14.5 KIPS/SQ.IN
DEFLECTION USED FOR CHOICE = 0.006  TOTAL LOAD USED IN DEFLEC. = 1.00  CASE = 2

PROPERTIES OF BEAMS USED

<table>
<thead>
<tr>
<th>SHAPE TYPE</th>
<th>ACI WT, LBS/FT</th>
<th>SEC. MODULUS, FT-LB/IN</th>
<th>MOM. INERTIA, FT-LB^2</th>
<th>INB</th>
<th>EFF. DEPTH, IN</th>
<th>D</th>
<th>WEB THICK, IN</th>
<th>ALLJR SPAN, FT</th>
<th>ALLJR CODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>6JR</td>
<td>6.40</td>
<td>2.40</td>
<td>7.30</td>
<td></td>
<td>6.00</td>
<td></td>
<td>0.114</td>
<td>13.00</td>
<td>1</td>
</tr>
<tr>
<td>7JR</td>
<td>7.50</td>
<td>2.80</td>
<td>12.00</td>
<td></td>
<td>7.00</td>
<td></td>
<td>0.120</td>
<td>15.00</td>
<td>1</td>
</tr>
<tr>
<td>8JR</td>
<td>8.60</td>
<td>3.20</td>
<td>16.70</td>
<td></td>
<td>8.00</td>
<td></td>
<td>0.125</td>
<td>15.00</td>
<td>1</td>
</tr>
<tr>
<td>10JR</td>
<td>9.40</td>
<td>3.60</td>
<td>18.10</td>
<td></td>
<td>10.00</td>
<td></td>
<td>0.130</td>
<td>15.00</td>
<td>2</td>
</tr>
<tr>
<td>11JR</td>
<td>10.40</td>
<td>4.00</td>
<td>85.80</td>
<td></td>
<td>12.00</td>
<td></td>
<td>0.135</td>
<td>17.00</td>
<td>2</td>
</tr>
<tr>
<td>12JR</td>
<td>11.40</td>
<td>4.40</td>
<td>51.90</td>
<td></td>
<td>12.00</td>
<td></td>
<td>0.140</td>
<td>17.00</td>
<td>2</td>
</tr>
<tr>
<td>13JR</td>
<td>12.40</td>
<td>4.80</td>
<td>72.70</td>
<td></td>
<td>15.00</td>
<td></td>
<td>0.145</td>
<td>25.00</td>
<td>1</td>
</tr>
<tr>
<td>12B</td>
<td>14.00</td>
<td>4.80</td>
<td>85.20</td>
<td></td>
<td>11.91</td>
<td></td>
<td>0.200</td>
<td>25.00</td>
<td>2</td>
</tr>
<tr>
<td>12B</td>
<td>16.50</td>
<td>7.50</td>
<td>105.30</td>
<td></td>
<td>12.00</td>
<td></td>
<td>0.230</td>
<td>25.00</td>
<td>1</td>
</tr>
<tr>
<td>1A</td>
<td>17.20</td>
<td>21.00</td>
<td>147.30</td>
<td></td>
<td>14.00</td>
<td></td>
<td>0.210</td>
<td>30.00</td>
<td>1</td>
</tr>
<tr>
<td>12A</td>
<td>19.00</td>
<td>21.40</td>
<td>160.30</td>
<td></td>
<td>12.16</td>
<td></td>
<td>0.220</td>
<td>30.00</td>
<td>2</td>
</tr>
<tr>
<td>10A</td>
<td>21.00</td>
<td>21.80</td>
<td>165.30</td>
<td></td>
<td>9.90</td>
<td></td>
<td>0.240</td>
<td>25.00</td>
<td>1</td>
</tr>
<tr>
<td>12A</td>
<td>22.00</td>
<td>25.30</td>
<td>155.70</td>
<td></td>
<td>12.91</td>
<td></td>
<td>0.260</td>
<td>25.00</td>
<td>1</td>
</tr>
<tr>
<td>12A</td>
<td>25.00</td>
<td>26.40</td>
<td>133.20</td>
<td></td>
<td>10.06</td>
<td></td>
<td>0.340</td>
<td>21.00</td>
<td>1</td>
</tr>
<tr>
<td>14A</td>
<td>22.00</td>
<td>28.80</td>
<td>197.40</td>
<td></td>
<td>13.72</td>
<td></td>
<td>0.230</td>
<td>30.00</td>
<td>1</td>
</tr>
<tr>
<td>14A</td>
<td>25.00</td>
<td>38.90</td>
<td>247.60</td>
<td></td>
<td>13.89</td>
<td></td>
<td>0.255</td>
<td>30.00</td>
<td>1</td>
</tr>
<tr>
<td>16A</td>
<td>26.00</td>
<td>38.10</td>
<td>295.10</td>
<td></td>
<td>15.65</td>
<td></td>
<td>0.250</td>
<td>34.00</td>
<td>1</td>
</tr>
<tr>
<td>16A</td>
<td>30.00</td>
<td>41.00</td>
<td>249.60</td>
<td></td>
<td>13.86</td>
<td></td>
<td>0.270</td>
<td>30.00</td>
<td>1</td>
</tr>
<tr>
<td>14A</td>
<td>31.00</td>
<td>47.00</td>
<td>327.50</td>
<td></td>
<td>13.86</td>
<td></td>
<td>0.270</td>
<td>34.00</td>
<td>1</td>
</tr>
<tr>
<td>14A</td>
<td>34.00</td>
<td>48.20</td>
<td>329.70</td>
<td></td>
<td>14.00</td>
<td></td>
<td>0.270</td>
<td>30.00</td>
<td>1</td>
</tr>
<tr>
<td>16A</td>
<td>36.00</td>
<td>56.30</td>
<td>466.30</td>
<td></td>
<td>13.85</td>
<td></td>
<td>0.299</td>
<td>34.00</td>
<td>1</td>
</tr>
<tr>
<td>16A</td>
<td>40.00</td>
<td>64.40</td>
<td>515.50</td>
<td></td>
<td>16.00</td>
<td></td>
<td>0.300</td>
<td>34.00</td>
<td>1</td>
</tr>
<tr>
<td>16A</td>
<td>45.00</td>
<td>72.40</td>
<td>553.30</td>
<td></td>
<td>16.12</td>
<td></td>
<td>0.306</td>
<td>34.00</td>
<td>1</td>
</tr>
<tr>
<td>16A</td>
<td>45.00</td>
<td>72.40</td>
<td>553.30</td>
<td></td>
<td>16.12</td>
<td></td>
<td>0.306</td>
<td>34.00</td>
<td>1</td>
</tr>
<tr>
<td>16A</td>
<td>50.00</td>
<td>80.70</td>
<td>655.40</td>
<td></td>
<td>16.28</td>
<td></td>
<td>0.320</td>
<td>34.00</td>
<td>1</td>
</tr>
<tr>
<td>18A</td>
<td>50.00</td>
<td>69.00</td>
<td>400.60</td>
<td></td>
<td>16.00</td>
<td></td>
<td>0.320</td>
<td>38.00</td>
<td>1</td>
</tr>
<tr>
<td>18A</td>
<td>55.00</td>
<td>83.20</td>
<td>838.90</td>
<td></td>
<td>14.21</td>
<td></td>
<td>0.300</td>
<td>38.00</td>
<td>1</td>
</tr>
<tr>
<td>18A</td>
<td>55.00</td>
<td>69.70</td>
<td>1169.70</td>
<td></td>
<td>20.80</td>
<td></td>
<td>0.320</td>
<td>43.00</td>
<td>1</td>
</tr>
<tr>
<td>24A</td>
<td>67.00</td>
<td>126.40</td>
<td>1329.80</td>
<td></td>
<td>20.99</td>
<td></td>
<td>0.400</td>
<td>43.00</td>
<td>1</td>
</tr>
<tr>
<td>SHAPE TYPE</td>
<td>ACW</td>
<td>JMAXx</td>
<td>JMAXz</td>
<td>INJ</td>
<td>ALLJ</td>
<td>ALLJ</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>------------</td>
<td>-----</td>
<td>-------</td>
<td>-------</td>
<td>-----</td>
<td>------</td>
<td>------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8H</td>
<td>4.20</td>
<td>73.00</td>
<td>200.00</td>
<td>11.00</td>
<td>16.00</td>
<td>13.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10H</td>
<td>4.20</td>
<td>91.00</td>
<td>220.00</td>
<td>18.00</td>
<td>14.00</td>
<td>16.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12H</td>
<td>4.10</td>
<td>118.00</td>
<td>250.00</td>
<td>22.00</td>
<td>20.00</td>
<td>20.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14H</td>
<td>6.10</td>
<td>148.00</td>
<td>280.00</td>
<td>27.00</td>
<td>20.00</td>
<td>16.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16H</td>
<td>4.50</td>
<td>111.00</td>
<td>240.00</td>
<td>26.00</td>
<td>24.00</td>
<td>20.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18H</td>
<td>5.70</td>
<td>140.00</td>
<td>280.00</td>
<td>33.00</td>
<td>24.00</td>
<td>20.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20H</td>
<td>6.70</td>
<td>180.00</td>
<td>320.00</td>
<td>40.00</td>
<td>28.00</td>
<td>20.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22H</td>
<td>7.10</td>
<td>222.00</td>
<td>360.00</td>
<td>49.00</td>
<td>24.00</td>
<td>20.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24H</td>
<td>8.20</td>
<td>260.00</td>
<td>400.00</td>
<td>56.00</td>
<td>24.00</td>
<td>20.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>26H</td>
<td>6.50</td>
<td>212.00</td>
<td>350.00</td>
<td>56.00</td>
<td>28.00</td>
<td>23.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>28H</td>
<td>7.40</td>
<td>259.60</td>
<td>380.00</td>
<td>65.00</td>
<td>26.00</td>
<td>23.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30H</td>
<td>8.60</td>
<td>307.00</td>
<td>420.00</td>
<td>81.00</td>
<td>29.00</td>
<td>23.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32H</td>
<td>10.00</td>
<td>359.00</td>
<td>460.00</td>
<td>93.00</td>
<td>28.00</td>
<td>23.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>34H</td>
<td>4.60</td>
<td>314.00</td>
<td>390.00</td>
<td>74.00</td>
<td>28.00</td>
<td>26.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>36H</td>
<td>7.60</td>
<td>369.00</td>
<td>430.00</td>
<td>90.00</td>
<td>32.00</td>
<td>26.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>38H</td>
<td>9.10</td>
<td>413.00</td>
<td>470.00</td>
<td>107.00</td>
<td>32.00</td>
<td>26.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>40H</td>
<td>11.50</td>
<td>459.00</td>
<td>510.00</td>
<td>124.00</td>
<td>32.00</td>
<td>26.00</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Determination of the most economical sections and spanning method for different orientations of steel beams and H series joists. Two main cases are considered which are structures for floors and structures for roof only.
Computation is made considering the most economical sections regardless of depth, which may also be entered as another constraint and can be handled by the same program.

Method of spanning.
Beams spanning length.
Joists spanning width.

<table>
<thead>
<tr>
<th>DAY SIZE BEAM SIZE</th>
<th>ACTUAL ALLOWED BEAM</th>
<th>DECK</th>
<th>JOIST SIZE</th>
<th>DECIMAL TOTAL COST</th>
</tr>
</thead>
<tbody>
<tr>
<td>FT X FT TYPE</td>
<td>WEIGHT</td>
<td>DEFLEC</td>
<td>DEFLEC</td>
<td>WEIGHT</td>
</tr>
<tr>
<td>12' X 12' 10H</td>
<td>11.50 0.307</td>
<td>0.96 0.19</td>
<td>10H2 4.20 0.147</td>
<td>2.10 0.29</td>
</tr>
<tr>
<td>12' X 14' 12H</td>
<td>14.00 0.336</td>
<td>1.17 0.23</td>
<td>10H2 4.20 0.147</td>
<td>2.10 0.29</td>
</tr>
<tr>
<td>12' X 16' 12H</td>
<td>16.50 0.307</td>
<td>1.38 0.28</td>
<td>10H2 4.20 0.147</td>
<td>2.10 0.29</td>
</tr>
<tr>
<td>12' X 18' 13H</td>
<td>17.20 0.253</td>
<td>1.43 0.29</td>
<td>10H2 4.20 0.147</td>
<td>2.10 0.29</td>
</tr>
<tr>
<td>12' X 20' 14H</td>
<td>22.00 0.632</td>
<td>1.63 0.37</td>
<td>10H2 4.20 0.147</td>
<td>2.10 0.29</td>
</tr>
<tr>
<td>12' X 22' 14H</td>
<td>26.00 0.615</td>
<td>2.17 0.43</td>
<td>10H2 4.20 0.147</td>
<td>2.10 0.29</td>
</tr>
<tr>
<td>12' X 24' 16H</td>
<td>26.00 0.871</td>
<td>2.17 0.43</td>
<td>10H2 4.20 0.147</td>
<td>2.10 0.29</td>
</tr>
<tr>
<td>12' X 26' 16H</td>
<td>31.00 0.953</td>
<td>2.58 0.52</td>
<td>10H2 4.20 0.147</td>
<td>2.10 0.29</td>
</tr>
<tr>
<td>12' X 28' 16H</td>
<td>36.00 1.009</td>
<td>3.00 0.60</td>
<td>10H2 4.20 0.147</td>
<td>2.10 0.29</td>
</tr>
<tr>
<td>12' X 30' 16H</td>
<td>40.00 1.068</td>
<td>3.38 0.67</td>
<td>10H2 4.20 0.147</td>
<td>2.10 0.29</td>
</tr>
<tr>
<td>12' X 32' 16H</td>
<td>45.00 1.194</td>
<td>3.75 0.75</td>
<td>10H2 4.20 0.147</td>
<td>2.10 0.29</td>
</tr>
<tr>
<td>12' X 34' 18H</td>
<td>45.00 1.321</td>
<td>3.75 0.75</td>
<td>10H2 4.20 0.147</td>
<td>2.10 0.29</td>
</tr>
<tr>
<td>12' X 36' 18H</td>
<td>50.00 1.683</td>
<td>4.17 0.83</td>
<td>10H2 4.20 0.147</td>
<td>2.10 0.29</td>
</tr>
<tr>
<td>12' X 38' 21H</td>
<td>55.00 1.666</td>
<td>4.58 0.92</td>
<td>10H2 4.20 0.147</td>
<td>2.10 0.29</td>
</tr>
<tr>
<td>12' X 40' 21H</td>
<td>65.00 1.700</td>
<td>4.58 0.92</td>
<td>10H2 4.20 0.147</td>
<td>2.10 0.29</td>
</tr>
<tr>
<td>12' X 42' 23H</td>
<td>70.00 1.794</td>
<td>5.18 0.99</td>
<td>10H2 4.20 0.147</td>
<td>2.10 0.29</td>
</tr>
<tr>
<td>12' X 44' 23H</td>
<td>75.00 1.800</td>
<td>5.18 0.99</td>
<td>10H2 4.20 0.147</td>
<td>2.10 0.29</td>
</tr>
<tr>
<td>12' X 46' 25H</td>
<td>85.00 1.949</td>
<td>5.93 0.99</td>
<td>10H2 4.20 0.147</td>
<td>2.10 0.29</td>
</tr>
<tr>
<td>12' X 48' 25H</td>
<td>90.00 1.977</td>
<td>5.93 0.99</td>
<td>10H2 4.20 0.147</td>
<td>2.10 0.29</td>
</tr>
<tr>
<td>12' X 50' 27H</td>
<td>95.00 2.111</td>
<td>6.43 0.99</td>
<td>10H2 4.20 0.147</td>
<td>2.10 0.29</td>
</tr>
<tr>
<td>12' X 52' 27H</td>
<td>102.00 2.111</td>
<td>6.43 0.99</td>
<td>10H2 4.20 0.147</td>
<td>2.10 0.29</td>
</tr>
<tr>
<td>12' X 54' 29H</td>
<td>112.00 2.192</td>
<td>7.19 0.99</td>
<td>10H2 4.20 0.147</td>
<td>2.10 0.29</td>
</tr>
<tr>
<td>12' X 56' 29H</td>
<td>117.00 2.192</td>
<td>7.19 0.99</td>
<td>10H2 4.20 0.147</td>
<td>2.10 0.29</td>
</tr>
<tr>
<td>12' X 58' 31H</td>
<td>127.00 2.378</td>
<td>7.81 0.99</td>
<td>10H2 4.20 0.147</td>
<td>2.10 0.29</td>
</tr>
<tr>
<td>12' X 60' 31H</td>
<td>132.00 2.378</td>
<td>7.81 0.99</td>
<td>10H2 4.20 0.147</td>
<td>2.10 0.29</td>
</tr>
</tbody>
</table>

Note: All sizes are in inches.
**DETERMINATION OF THE MOST ECONOMICAL SECTIONS AND SPANNING METHOD FOR DIFFERENT BAY SIZES AND USING STEEL BEAMS AND H SERIES JOISTS.**

Two main cases are considered which are structures for floors and structures for roof only. Computation is made considering the most economical sections regardless of depth, which may also be entered as another constraint and can be handled by the same program.

**METHOD OF SPANNING.**

Beams spanning width, Joists spanning length.

<table>
<thead>
<tr>
<th>BAY SIZE</th>
<th>BEAM FT X FT</th>
<th>BEAM SIZE</th>
<th>BEAM ACTUAL ALLOWED</th>
<th>BEAM DEC.</th>
<th>JOIST SIZE</th>
<th>JOIST ACTUAL ALLOWED</th>
<th>JOIST DEC.</th>
<th>TOTAL COST</th>
</tr>
</thead>
<tbody>
<tr>
<td>(W)</td>
<td>(L)</td>
<td>WEIGHT</td>
<td>DEFLEC</td>
<td>DEFLEC</td>
<td>WEIGHT</td>
<td>COST</td>
<td>STN</td>
<td>WEIGHT</td>
</tr>
<tr>
<td>FT</td>
<td>FT</td>
<td>LBS/FT</td>
<td>INCHES</td>
<td>INCHES</td>
<td>LBS/SFT</td>
<td>$/SOFT</td>
<td>RULE</td>
<td>LBS/FT</td>
</tr>
<tr>
<td>12, x 12, 10n</td>
<td>10n</td>
<td>11.50</td>
<td>0.307</td>
<td>0.96</td>
<td>0.19</td>
<td>10H2</td>
<td>4.20</td>
<td>0.147</td>
</tr>
<tr>
<td>12, x 14, 10n</td>
<td>10n</td>
<td>11.50</td>
<td>0.358</td>
<td>0.82</td>
<td>0.16</td>
<td>10H2</td>
<td>4.20</td>
<td>0.272</td>
</tr>
<tr>
<td>12, x 16, 12n</td>
<td>12n</td>
<td>14.00</td>
<td>0.241</td>
<td>0.88</td>
<td>0.14</td>
<td>10H2</td>
<td>4.20</td>
<td>0.448</td>
</tr>
<tr>
<td>12, x 18, 12n</td>
<td>12n</td>
<td>14.00</td>
<td>0.271</td>
<td>0.82</td>
<td>0.21</td>
<td>10H2</td>
<td>4.50</td>
<td>0.515</td>
</tr>
<tr>
<td>12, x 20, 12n</td>
<td>12n</td>
<td>16.50</td>
<td>0.253</td>
<td>0.83</td>
<td>0.17</td>
<td>12H2</td>
<td>4.20</td>
<td>0.785</td>
</tr>
<tr>
<td>12, x 22, 12n</td>
<td>12n</td>
<td>16.50</td>
<td>0.279</td>
<td>0.75</td>
<td>0.15</td>
<td>14H3</td>
<td>5.50</td>
<td>0.668</td>
</tr>
<tr>
<td>12, x 24, 14n</td>
<td>14n</td>
<td>17.20</td>
<td>0.219</td>
<td>0.72</td>
<td>0.14</td>
<td>16H4</td>
<td>6.50</td>
<td>0.594</td>
</tr>
<tr>
<td>12, x 26, 14n</td>
<td>14n</td>
<td>17.20</td>
<td>0.237</td>
<td>0.66</td>
<td>0.13</td>
<td>16H4</td>
<td>6.60</td>
<td>0.805</td>
</tr>
<tr>
<td>12, x 28, 14n</td>
<td>14n</td>
<td>19.00</td>
<td>0.201</td>
<td>0.68</td>
<td>0.15</td>
<td>18H5</td>
<td>8.00</td>
<td>0.697</td>
</tr>
<tr>
<td>12, x 30, 14n</td>
<td>14n</td>
<td>22.00</td>
<td>0.206</td>
<td>0.69</td>
<td>0.15</td>
<td>20H6</td>
<td>8.00</td>
<td>0.918</td>
</tr>
<tr>
<td>12, x 32, 14n</td>
<td>14n</td>
<td>22.00</td>
<td>0.220</td>
<td>0.69</td>
<td>0.15</td>
<td>20H6</td>
<td>8.00</td>
<td>0.956</td>
</tr>
<tr>
<td>12, x 34, 14n</td>
<td>14n</td>
<td>22.00</td>
<td>0.235</td>
<td>0.65</td>
<td>0.13</td>
<td>22H6</td>
<td>9.79</td>
<td>0.846</td>
</tr>
<tr>
<td>12, x 36, 14n</td>
<td>14n</td>
<td>22.00</td>
<td>0.248</td>
<td>0.61</td>
<td>0.12</td>
<td>22H6</td>
<td>9.79</td>
<td>1.043</td>
</tr>
<tr>
<td>12, x 38, 16n</td>
<td>16n</td>
<td>26.00</td>
<td>0.174</td>
<td>0.68</td>
<td>0.15</td>
<td>24H6</td>
<td>10.30</td>
<td>1.106</td>
</tr>
<tr>
<td>12, x 40, 16n</td>
<td>16n</td>
<td>26.00</td>
<td>0.183</td>
<td>0.65</td>
<td>0.13</td>
<td>24H6</td>
<td>10.30</td>
<td>1.354</td>
</tr>
<tr>
<td>12, x 42, 16n</td>
<td>16n</td>
<td>26.00</td>
<td>0.228</td>
<td>1.10</td>
<td>0.30</td>
<td>26H7</td>
<td>12.00</td>
<td>1.444</td>
</tr>
<tr>
<td>12, x 44, 16n</td>
<td>16n</td>
<td>26.00</td>
<td>0.312</td>
<td>1.03</td>
<td>0.21</td>
<td>26H7</td>
<td>12.00</td>
<td>2.082</td>
</tr>
<tr>
<td>12, x 46, 16n</td>
<td>16n</td>
<td>26.00</td>
<td>0.375</td>
<td>0.96</td>
<td>0.19</td>
<td>28H8</td>
<td>12.00</td>
<td>2.515</td>
</tr>
<tr>
<td>12, x 48, 16n</td>
<td>16n</td>
<td>26.00</td>
<td>0.395</td>
<td>0.96</td>
<td>0.19</td>
<td>28H8</td>
<td>12.00</td>
<td>3.018</td>
</tr>
<tr>
<td>12, x 50, 16n</td>
<td>16n</td>
<td>26.00</td>
<td>0.437</td>
<td>0.89</td>
<td>0.17</td>
<td>30H9</td>
<td>12.00</td>
<td>3.384</td>
</tr>
<tr>
<td>12, x 52, 16n</td>
<td>16n</td>
<td>26.00</td>
<td>0.473</td>
<td>0.89</td>
<td>0.17</td>
<td>30H9</td>
<td>12.00</td>
<td>4.228</td>
</tr>
<tr>
<td>12, x 54, 16n</td>
<td>16n</td>
<td>26.00</td>
<td>0.508</td>
<td>0.82</td>
<td>0.16</td>
<td>32H10</td>
<td>12.00</td>
<td>4.876</td>
</tr>
<tr>
<td>12, x 56, 16n</td>
<td>16n</td>
<td>26.00</td>
<td>0.539</td>
<td>0.82</td>
<td>0.16</td>
<td>32H10</td>
<td>12.00</td>
<td>5.524</td>
</tr>
</tbody>
</table>
IN ORDER TO HAVE A COMPLETE IMAGE ABOUT THE RESULTS LET'S PLOT THE RESULTS IN A GRAPH OF COST VS BAY SIZE.

SINCE VERTICAL AXIS HAS BEEN LIMITED TO A MAXIMUM OF 55 LINES AN INDEX HAS BEEN COMPUTED IN ORDER TO OBTAIN THE ACTUAL FIGURES OF COST VERTICAL SCALE SHOWS ACTUAL COST, ACCORDING TO SUCH AN INDEX, THEREFORE THEY ARE AN APPROXIMATION TO THE REAL COST, GIVEN IN TABLES.

FOR THE CASE OF BEAMS SPANNING LENGTH AND JOISTS SPANNING WIDTH THE SYMBOL USED IN THE GRAPH IS (X)

FOR THE CASE OF BEAMS SPANNING WIDTH AND JOISTS SPANNING LENGTH THE SYMBOL USED IN THE GRAPH IS (H)

FOR SQUARE BAYS AND EQUAL COSTS THE SYMBOL USED IN THE GRAPH IS (O)
<table>
<thead>
<tr>
<th>Size (FT)</th>
<th>Cost per Square Foot</th>
<th>Bay Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>14</td>
<td>16</td>
</tr>
<tr>
<td>12</td>
<td>14</td>
<td>18</td>
</tr>
<tr>
<td>12</td>
<td>14</td>
<td>20</td>
</tr>
</tbody>
</table>

COST PER SQUARE FOOT VS BAY SIZE

GRAPH OF TOTAL COST FOR BOTH SPANNING METHODS.
DETERMINATION OF THE MOST ECONOMICAL SECTIONS AND SPANNING METHOD FOR DIFFERENT BAY SIZES AND USING STEEL BEAMS AND H SERIES JOISTS.

TWO MAIN CASES ARE CONSIDERED WHICH ARE STRUCTURES FOR FLOORS AND STRUCTURES FOR ROOF ONLY.

COMPUTATION IS MADE CONSIDERING THE MOST ECONOMICAL SECTIONS REGARDLESS OF DEPTH, WHICH MAY ALSO BE ENTERED AS ANOTHER CONSTRAINT AND CAN BE HANDLED BY THE SAME PROGRAM.

### Method of Spanning

**Heads Spanning Length:**

**Joists Spanning Width:**

<table>
<thead>
<tr>
<th>BAY SIZE</th>
<th>BEAM SIZE</th>
<th>BEAM TYPE</th>
<th>S/F</th>
<th>ACTUAL</th>
<th>ALLOWED</th>
<th>BEAM WEIGHT</th>
<th>DECISION RULE</th>
<th>JOIST SIZE</th>
<th>ACTUAL</th>
<th>ALLOWED</th>
<th>JOIST WEIGHT</th>
<th>DECISION RULE</th>
<th>TOTAL COST</th>
</tr>
</thead>
<tbody>
<tr>
<td>12' x 12'</td>
<td>10H</td>
<td>14.00</td>
<td>0.307</td>
<td>0.440</td>
<td>0.96</td>
<td>0.19</td>
<td>MONT</td>
<td>10H</td>
<td>4.20</td>
<td>0.147</td>
<td>0.400</td>
<td>2.10</td>
<td>0.29</td>
</tr>
<tr>
<td>12' x 14'</td>
<td>12H</td>
<td>14.00</td>
<td>0.336</td>
<td>0.467</td>
<td>1.17</td>
<td>0.23</td>
<td>MONT</td>
<td>10H</td>
<td>4.20</td>
<td>0.147</td>
<td>0.400</td>
<td>2.10</td>
<td>0.29</td>
</tr>
<tr>
<td>12' x 16'</td>
<td>16H</td>
<td>16.39</td>
<td>0.422</td>
<td>0.533</td>
<td>1.38</td>
<td>0.28</td>
<td>MONT</td>
<td>10H</td>
<td>4.20</td>
<td>0.147</td>
<td>0.400</td>
<td>2.10</td>
<td>0.29</td>
</tr>
<tr>
<td>12' x 18'</td>
<td>18H</td>
<td>17.60</td>
<td>0.531</td>
<td>0.600</td>
<td>1.43</td>
<td>0.31</td>
<td>MONT</td>
<td>10H</td>
<td>4.20</td>
<td>0.147</td>
<td>0.400</td>
<td>2.10</td>
<td>0.29</td>
</tr>
<tr>
<td>12' x 20'</td>
<td>20H</td>
<td>22.00</td>
<td>0.632</td>
<td>0.667</td>
<td>1.63</td>
<td>0.37</td>
<td>BFE</td>
<td>10H</td>
<td>4.20</td>
<td>0.147</td>
<td>0.400</td>
<td>2.10</td>
<td>0.29</td>
</tr>
<tr>
<td>12' x 22'</td>
<td>22H</td>
<td>21.60</td>
<td>0.615</td>
<td>0.733</td>
<td>2.17</td>
<td>0.43</td>
<td>BFE</td>
<td>10H</td>
<td>4.20</td>
<td>0.147</td>
<td>0.400</td>
<td>2.10</td>
<td>0.29</td>
</tr>
<tr>
<td>12' x 24'</td>
<td>24H</td>
<td>31.00</td>
<td>0.701</td>
<td>0.800</td>
<td>2.58</td>
<td>0.32</td>
<td>BFE</td>
<td>10H</td>
<td>4.20</td>
<td>0.147</td>
<td>0.400</td>
<td>2.10</td>
<td>0.29</td>
</tr>
<tr>
<td>12' x 26'</td>
<td>26H</td>
<td>33.00</td>
<td>0.710</td>
<td>0.875</td>
<td>2.75</td>
<td>0.39</td>
<td>BFE</td>
<td>10H</td>
<td>4.20</td>
<td>0.147</td>
<td>0.400</td>
<td>2.10</td>
<td>0.29</td>
</tr>
<tr>
<td>12' x 28'</td>
<td>28H</td>
<td>37.00</td>
<td>0.725</td>
<td>0.925</td>
<td>2.75</td>
<td>0.43</td>
<td>BFE</td>
<td>10H</td>
<td>4.20</td>
<td>0.147</td>
<td>0.400</td>
<td>2.10</td>
<td>0.29</td>
</tr>
<tr>
<td>12' x 30'</td>
<td>30H</td>
<td>42.00</td>
<td>1.050</td>
<td>1.067</td>
<td>4.17</td>
<td>0.43</td>
<td>BFE</td>
<td>10H</td>
<td>4.20</td>
<td>0.147</td>
<td>0.400</td>
<td>2.10</td>
<td>0.29</td>
</tr>
<tr>
<td>12' x 32'</td>
<td>32H</td>
<td>47.00</td>
<td>1.151</td>
<td>1.333</td>
<td>5.67</td>
<td>1.13</td>
<td>BFE</td>
<td>10H</td>
<td>4.20</td>
<td>0.147</td>
<td>0.400</td>
<td>2.10</td>
<td>0.29</td>
</tr>
<tr>
<td>12' x 34'</td>
<td>34H</td>
<td>52.00</td>
<td>1.267</td>
<td>1.667</td>
<td>5.67</td>
<td>1.13</td>
<td>BFE</td>
<td>10H</td>
<td>4.20</td>
<td>0.147</td>
<td>0.400</td>
<td>2.10</td>
<td>0.29</td>
</tr>
<tr>
<td>12' x 36'</td>
<td>36H</td>
<td>70.00</td>
<td>1.402</td>
<td>1.937</td>
<td>5.95</td>
<td>1.16</td>
<td>BFE</td>
<td>10H</td>
<td>4.20</td>
<td>0.147</td>
<td>0.400</td>
<td>2.10</td>
<td>0.29</td>
</tr>
<tr>
<td>12' x 38'</td>
<td>38H</td>
<td>79.00</td>
<td>1.478</td>
<td>1.967</td>
<td>6.00</td>
<td>1.19</td>
<td>BFE</td>
<td>10H</td>
<td>4.20</td>
<td>0.147</td>
<td>0.400</td>
<td>2.10</td>
<td>0.29</td>
</tr>
<tr>
<td>12' x 40'</td>
<td>40H</td>
<td>99.00</td>
<td>1.667</td>
<td>2.167</td>
<td>6.00</td>
<td>1.19</td>
<td>BFE</td>
<td>10H</td>
<td>4.20</td>
<td>0.147</td>
<td>0.400</td>
<td>2.10</td>
<td>0.29</td>
</tr>
<tr>
<td>12' x 42'</td>
<td>42H</td>
<td>119.00</td>
<td>1.833</td>
<td>2.333</td>
<td>6.00</td>
<td>1.19</td>
<td>BFE</td>
<td>10H</td>
<td>4.20</td>
<td>0.147</td>
<td>0.400</td>
<td>2.10</td>
<td>0.29</td>
</tr>
<tr>
<td>12' x 44'</td>
<td>44H</td>
<td>139.00</td>
<td>1.992</td>
<td>2.333</td>
<td>6.00</td>
<td>1.19</td>
<td>BFE</td>
<td>10H</td>
<td>4.20</td>
<td>0.147</td>
<td>0.400</td>
<td>2.10</td>
<td>0.29</td>
</tr>
<tr>
<td>12' x 46'</td>
<td>46H</td>
<td>159.00</td>
<td>2.151</td>
<td>2.333</td>
<td>6.00</td>
<td>1.19</td>
<td>BFE</td>
<td>10H</td>
<td>4.20</td>
<td>0.147</td>
<td>0.400</td>
<td>2.10</td>
<td>0.29</td>
</tr>
<tr>
<td>12' x 48'</td>
<td>48H</td>
<td>179.00</td>
<td>2.300</td>
<td>2.333</td>
<td>6.00</td>
<td>1.19</td>
<td>BFE</td>
<td>10H</td>
<td>4.20</td>
<td>0.147</td>
<td>0.400</td>
<td>2.10</td>
<td>0.29</td>
</tr>
</tbody>
</table>

### Method of Spanning

**Heads Spanning Length:**

**Joists Spanning Width:**
### Determination of the Most Economical Sections and Spanning Method for Different Day Sizes and Using Steel Beams and H Series Joists

This main case is considered which are structures for flouors and structures for roof only. Computation is made concerning the most economical sections regardless of depth, which may also be entered as another constraint and can be handled by the same program.

### Method of Spanning

<table>
<thead>
<tr>
<th>Beam Spacing</th>
<th>Joist Spanning Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>(W) X (L)</td>
<td>FT</td>
</tr>
</tbody>
</table>

### Beam Size and Heam Type

<table>
<thead>
<tr>
<th>Size (W) X (L)</th>
<th>Type</th>
<th>Weight</th>
<th>Deflec</th>
<th>Deflec</th>
<th>Weight</th>
<th>Beam Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>12 X 12</td>
<td>10a</td>
<td>11.50</td>
<td>0.307</td>
<td>0.400</td>
<td>0.495</td>
<td>0.19</td>
</tr>
<tr>
<td>12 X 12</td>
<td>10d</td>
<td>11.50</td>
<td>0.307</td>
<td>0.400</td>
<td>0.495</td>
<td>0.19</td>
</tr>
</tbody>
</table>

### Deciding Joist Size and Actual Deflection

<table>
<thead>
<tr>
<th>Joist Size</th>
<th>Actual Deflection</th>
<th>Allowand Joist</th>
<th>Joist Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>12 X 12</td>
<td>4.20</td>
<td>0.307</td>
<td>0.437</td>
</tr>
</tbody>
</table>

### Beam Span and Total Cost

<table>
<thead>
<tr>
<th>Span Rule</th>
<th>Dlls/Soft</th>
</tr>
</thead>
<tbody>
<tr>
<td>12 X 12</td>
<td>4.90</td>
</tr>
</tbody>
</table>

---

### Table for Different Day Sizes

<table>
<thead>
<tr>
<th>Day Sizes (W) X (L)</th>
<th>Type</th>
<th>Weight</th>
<th>Deflec</th>
<th>Deflec</th>
<th>Weight</th>
<th>Beam Cost</th>
<th>Span Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>12 X 12</td>
<td>10a</td>
<td>11.50</td>
<td>0.307</td>
<td>0.400</td>
<td>0.495</td>
<td>0.19</td>
<td>4.90</td>
</tr>
<tr>
<td>12 X 12</td>
<td>10d</td>
<td>11.50</td>
<td>0.307</td>
<td>0.400</td>
<td>0.495</td>
<td>0.19</td>
<td>4.90</td>
</tr>
</tbody>
</table>
In order to have a complete image about the results let's plot the results in a graph of cost vs. ray size.

Since vertical axis has been limited to a maximum of 5k lines an index has been computed in order to obtain the actual figures of cost. Vertical scale shows actual cost, according to such an index, therefore they are an approximation to the real cost, given in tables.

For the case of beams spanning length and joists spanning width, the symbol used in the graph is (H). For the case of beams spanning width and joists spanning length, the symbol used in the graph is (X). For square rays and equal costs the symbol used in the graph is (O).
### Determination of the Most Economical Sections and Spanning Method for Different Bay Sizes and Using Steel Beams and H Series Joists

Two main cases are considered which are structures for floors and structures for roof only. Computation is 'made considering the most economical sections regardless of depth, which may also be entered as another constraint and can be handled by the same program.'

---

**Method of Spanning:**

- Beams spanning length
- Joists spanning width

<table>
<thead>
<tr>
<th>Bay</th>
<th>Beam Size</th>
<th>Height Deflec</th>
<th>Deflec</th>
<th>Beam Cost</th>
<th>Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>FT</td>
<td>FT</td>
<td>FT/Lbs</td>
<td>$/Lb</td>
<td>FT/Soft</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>10</td>
<td>0.96</td>
<td>0.19</td>
<td>0.00</td>
<td>Rule</td>
</tr>
<tr>
<td>12</td>
<td>12</td>
<td>0.86</td>
<td>0.06</td>
<td>0.00</td>
<td>Rule</td>
</tr>
<tr>
<td>12</td>
<td>14</td>
<td>0.76</td>
<td>0.06</td>
<td>0.00</td>
<td>Rule</td>
</tr>
<tr>
<td>12</td>
<td>16</td>
<td>0.69</td>
<td>0.06</td>
<td>0.00</td>
<td>Rule</td>
</tr>
</tbody>
</table>

No section available was found for shell case.
DETERMINATION OF THE MOST ECONOMICAL SECTIONS AND SPANNING METHOD FOR DIFFERENT BAY SIZES AND USING STEEL BEAMS AND H SERIES JOISTS.

TWO MAIN CASES ARE CONSIDERED WHICH ARE STRUCTURES FOR FLOORS AND STRUCTURES FOR ROOF ONLY. COMPUTATION IS MADE CONSIDERING THE MOST ECONOMICAL SECTIONS REGARDLESS OF DEPTH, WHICH MAY ALSO BE ENTERED AS ANOTHER CONSTRAINT AND CAN BE HANDLED BY THE SAME PROGRAM.

**Method of Spanning.**
BEAMS SPANNING WIDTH,
JOISTS SPANNING LENGTH.

<table>
<thead>
<tr>
<th>BAY SIZE</th>
<th>BEAM SIZE</th>
<th>ACTUAL ALLOWED BEAM DECISION</th>
</tr>
</thead>
<tbody>
<tr>
<td>(W) X (L)</td>
<td>TYPE</td>
<td>WEIGHT</td>
</tr>
<tr>
<td>FT</td>
<td>FT</td>
<td>LBS/FT</td>
</tr>
<tr>
<td>12</td>
<td>12</td>
<td>10H</td>
</tr>
<tr>
<td>12</td>
<td>16</td>
<td>12H</td>
</tr>
<tr>
<td>12</td>
<td>20</td>
<td>14H</td>
</tr>
<tr>
<td>12</td>
<td>24</td>
<td>14H</td>
</tr>
<tr>
<td>12</td>
<td>32</td>
<td>14H</td>
</tr>
<tr>
<td>12</td>
<td>36</td>
<td>NONE</td>
</tr>
<tr>
<td>12</td>
<td>40</td>
<td>NONE</td>
</tr>
<tr>
<td>16</td>
<td>16</td>
<td>12H</td>
</tr>
<tr>
<td>16</td>
<td>20</td>
<td>14H</td>
</tr>
<tr>
<td>16</td>
<td>24</td>
<td>16H</td>
</tr>
<tr>
<td>16</td>
<td>32</td>
<td>16H</td>
</tr>
<tr>
<td>16</td>
<td>36</td>
<td>NONE</td>
</tr>
<tr>
<td>20</td>
<td>20</td>
<td>16H</td>
</tr>
<tr>
<td>20</td>
<td>24</td>
<td>18H</td>
</tr>
<tr>
<td>20</td>
<td>32</td>
<td>16H</td>
</tr>
<tr>
<td>24</td>
<td>24</td>
<td>20H</td>
</tr>
<tr>
<td>24</td>
<td>32</td>
<td>21H</td>
</tr>
<tr>
<td>28</td>
<td>36</td>
<td>NONE</td>
</tr>
<tr>
<td>28</td>
<td>40</td>
<td>NONE</td>
</tr>
<tr>
<td>32</td>
<td>28</td>
<td>21H</td>
</tr>
<tr>
<td>32</td>
<td>32</td>
<td>NONE</td>
</tr>
<tr>
<td>Section</td>
<td>Numeral</td>
<td>Numeral</td>
</tr>
<tr>
<td>---------</td>
<td>---------</td>
<td>---------</td>
</tr>
<tr>
<td>28 x 36</td>
<td>None</td>
<td>0.00</td>
</tr>
<tr>
<td>28 x 40</td>
<td>None</td>
<td>0.00</td>
</tr>
<tr>
<td>32 x 36</td>
<td>29.6</td>
<td>0.00</td>
</tr>
<tr>
<td>32 x 40</td>
<td>None</td>
<td>0.00</td>
</tr>
<tr>
<td>36 x 36</td>
<td>None</td>
<td>0.00</td>
</tr>
<tr>
<td>36 x 40</td>
<td>None</td>
<td>0.00</td>
</tr>
<tr>
<td>40 x 40</td>
<td>None</td>
<td>0.00</td>
</tr>
</tbody>
</table>
SUBROUTINE PERFS(ARRI, ARRI, SCAL, CYCLE, LINE, DOT, BLANK, STAR, ITAL, R 0000
I TER, IN, WD, DIST, IJ, MIN, MAX, INDEX) R 0000
INTEGER ARRI(I), ARRI(I), LINE(J), SCAL, CYCLE, ND R 0000
REAL INDEX R 0000
WRITE (6,4500) R 0000
WRITE (6,2000) R 0003
WRITE (6,5000) R 0006
WRITE (6,5010) R 0009
WRITE (6,5020) INDEX R 0012
WRITE (6,5030) R 0021
WRITE (6,5040) R 0024
WRITE (6,5050) R 0027
WRITE (6,2000) R 0030
MIN = (MIN/INDEX) + 0.5 R 0033
MAX = (MAX/INDEX) + 0.5 R 0037
MIN2 = MIN - N R 0041
ND = (ND = DIST)/DIST R 0045
NI = IN R 0046
IV = 10 R 0047
IVEN = 10 R 0047
ND = 0 JX = 10 CYCLE R 0047
GO TO 200 R 0052
IF (JX > oo) GO TO 200 R 0055
IF (JX > N1) 220, 200, 220 R 0059
200 ND 210 L = 10 SCAL R 0064
LINE(L) = DOT R 0068
210 CONTINUE R 0070
GO TO 240 R 0074
220 ND 230 L = 10 SCAL R 0075
LINE(L) = BLANK R 0079
230 CONTINUE R 0080
KPC = ARRI(JX) = MIN R 0082
KPC = ARRI(JX) = MIN R 0085
IF (ARRI(JX) = ARRI(JX)) 240, 200, 220 R 0087
240 LKE(KPC) = ZER R 0094
GO TO 220 R 0095
250 ND 260 JHE = KPC*KPC R 0097
LINE(JHE) = STAR R 0102
265 CONTINUE R 0106
GO TO 300 R 0108
270 ND 275 JHE = KPC*KPC R 0108
LINE(JHE) = ITAL R 0113
275 CONTINUE R 0116
280 IVER = IVER + 2 R 0119
IF (JX > 100) 280, 200, 220 R 0121
282 WRITE (6,5050) IVER, IVER, LINE(KP), KP = 10 SCAL R 0124
GO TO 300 R 0125
280 ND 280 JHE = KPC*KPC R 0142
WRITE (6,5050) IVER, IVER, LINE(KP), KP = 10 SCAL R 0146
GO TO 300 R 0146
290 IV = IV + 2 R 0148
IVER = IV R 0168
WRITE (6,5050) IVER, IVER, LINE(KP), KP = 10 SCAL R 0170
IV = IV + 2 R 0170
IVER = IV R 0170
WRITE (6,5050) MIN, MAX R 0179
WRITE (6,5050) R 0206
C FORMAT STATEMENTS, R 0208
2000 FORMAT (1H1) R 0209
4500 FORMAT (1H1) R 0209
5000 FORMAT (9X,13",O*",","110A1",")
5005 FORMAT (9X,"IN ORDER TO HAVE A COMPLETE IMAGE ABOUT")
1 THE RESULTS LET'S PLOT THE",9X,"RESULTS IN A GRAPH OF COST VS.
2 RAY SIZE.")
5010 FORMAT (9X,"SINCE HORIZONTAL AXIS HAS BEEN LIMITED TO A MAXI")
1UM OF 110 COLUMNS",/9X,"A SCALE FACTOR HAS BEEN COMPUTED AND IN 2 ORDER TO OBTAIN")
2 ACTUAL FIGURES",9X,"OF COST IT IS NECESSARY TO")
3 MULTIPLY THE FORMAL HORIZONTAL SCALE TIMES",9X,"THE SCALE FACTOR")
4 DEFINED BELOW.")

SEGMENT A.15 123.103 LONG

5020 FORMAT (9X,"THE INDEX FOR THE NEXT GRAPH IS ",9X,"84XE15.7")
5030 FORMAT (9X,"FOR THE CASE IN WHICH UPPER COST CORRESPONDS TO")
1REAMS SPANNING LENGTH",9X,"AND JOISTS SPANNING WIDTH, THE GRAPH")
2 WILL HAVE SYMBOL",9X,"(X)"
5040 FORMAT (9X,"FOR THE CASE IN WHICH UPPER COST CORRESPONDS TO REA")
1YS SPANNING WIDTH",9X,"AND JOISTS SPANNING LENGTH, THE GRAPH WILL")
2 HAVE SYMBOL",9X,"(X)"
5050 FORMAT (9X,"FOR SQUARE RAYS AND EQUAL COSTS THE SYMBOL USED IN")
1THE GRAPH IS ",9X,"(X)"
5500 FORMAT (9X,"",11X",0","11X",0","110A1",")

SEGMENT A.15 245.103 LONG

6500 FORMAT (16X,"",0","102X","",0")
7000 FORMAT (",",1X,"HORIZONTAL SCALE OF COSTS, WITH LOWER AND UPPER COS")
ITS, TO READ USE THE SCALE FACTOR,")
RETURN
END

SEGMENT A.15 245.103 LONG
In order to have a complete image about the results let's plot the results in a graph of cost vs ray size.

Since horizontal axis has been limited to a maximum of 110 columns, a scale factor has been computed and in order to obtain actual figures of cost it is necessary to multiply the formal horizontal scale times the scale factor defined below.

The index for the next graph is \(1.000000001\).

For the case in which upper cost corresponds to beams spanning length and joists spanning width, the graph will have symbol \(\times\).

For the case in which upper cost corresponds to beams spanning width and joists spanning length, the graph will have symbol \(\cdot\).

For square rays and equal costs the symbol used in the graph is \(\circ\).
END PUNCH 3.4 SEC.

LOAD GATOS-
FILE: GATOS -- TYPE: FORTRAN -- LOADING
END LOAD 1.7 SEC.

RUN-
COMPILING.

END COMPILE 12.6 SEC.

RUNNING

GRAPH OF COSTS VS BAY SIZE
SCALE FACTOR FOR GRAPH = 1.7813
BEAMS SPAN LENGTH UPPER COST SYMBOL = H
BEAMS SPAN WIDTH UPPER COST SYMBOL = X
EQUAL COST AND SQUARE BAYS SYMBOL = O

12.0 X 12.0 0 0
14.0 *HHHHH
16.0 *HHHHHH
18.0 *HHHHHHH
20.0 * HH
22.0 * HH
24.0 * HH
26.0 * HH
28.0 * HH
30.0 * HH
32.0 * HH
34.0 * HH
36.0 * HH
38.0 * HH
40.0 * HH

14.0 X 14.0 0 0
16.0 * HH
18.0 * HH
20.0 * HH
22.0 * HH
24.0 * HH
26.0 * HH
28.0 * HH
30.0 * HH
32.0 * HH
34.0 * HH
36.0 * HH
38.0 * HH
40.0 * HH

16.0 X 16.0 0 0
18.0 * HH
HORIZONTAL SCALE OF COSTS
INCREASE EACH POINT BY ONE UNIT, AND MULTIPLY TIMES SCALE FACTOR TO
\ \ GET COST

END GATOS 19.6 SEC.

BYE-
ON FOR 58 MIN, 49.4 SEC.
CAE USE 3.9 SEC.
EXECUTE 50.4 SEC.
OFF AT 6:13 PM.
GOODBYE 1300020
04/01/70
SUBROUTINE COPLOT (A,B1,H2,SC,CYCA,WD,N,MIN,MAX,IND,VSCALE,  
SYM,HSCALE,TITL1,TITL2,TITL3,BUF,N1,N2,N3,N4,N5,RP1,RP2,IFIG6)    
INTEGER SC,CYCA,W  
DIMENSION A(CYCA),H1(CYCA),H2(CYCA),IFIG6(2)  
INTEGER H1(CYCA),H2(CYCA),IFIG6(1),VSCALE(N1),SYM(NM)  
INTEGER HSCALE,N3,N4,N5,N1,N2,N3,N4,N5,N1,N2  
REAL IND  
C  
MAX = MIN/IND + 0.5  
MIN = MAX/IND + 0.5  
C ASSIGN VALUES TO REAL AXIS X COORDINATES FOR A 0.20 UNIT INCREMENT.  
C I.E., FIVE POINTS PER INCH.  
DO 100 JK = 1,CYCA  
AC(JK) = (JK-1)/5.0  
100 CONTINUE  
C ASSIGN VALUES TO REAL AXIS Y COORDINATES FOR A 0.20 UNIT INCREMENT.  
C I.E., FIVE POINTS PER INCH.  
DO 150 JY = 1,CYCA  
BY(JY) = (JY(JY)-MIN)/5.0  
150 CONTINUE  
X1 = -0.10  
X2 = -1.00  
X3 = -1.10  
C START CALLING SEQUENCE AND DEFINE LOGICAL ORIGIN.  
CALL PLOT (IIND+1)  
CALL PLOT (0,0,0,0,0,-1)  
C SET CYCLE FOR DRAWING VERTICAL AXIS.  
DO 200 JY=1,SC+1  
MIN = (MIN*IND) + 0.5  
CPR = MIN/IND+0.0  
200 CONTINUE  
C DRAW TICK MARKS FOR VERTICAL AXIS.  
CALL NUMBER (Y1,Y1,Y1,Y1,Y1)  
C DRAW TICK MARKS FOR SCALE REFERENCE.  
CALL PLOT (X1,Y1+3)  
CALL PLOT (X1,Y1+3)  
Y1 = Y1 + 0.20  
Y2 = Y1 + 0.20  
MIN = MIN + 1  
C DRAW VERTICAL AXIS TITLE.  
CALL SYMBOL (01,02,0,0,0,0,0,0,20,VSCEA(1),90,0,20)  
C DRAW PERIMETER OF GRAPH.  
CALL PLOT (X1+P2,P2)  
CALL PLOT (P3+P2,P2)  
CALL PLOT (P3+P2,P2)  
CALL PLOT (P1,P1)  
Y1 = Y1 - 0.0
C SET CYCLE FOR DRAWING HORIZONTAL AXIS AND VERTICAL REFERENCE LINES.

ON 1 100 JK = 1 CYC
a = VY = VY + 2.0
IF (JK = 0.1) GO TO 525
IF (JK = 0.1) GO TO 500

C DRAW HORIZONTAL BAY SIZE FIGURES FOR NUN SQUARE BAYS.

CALL NUMBER (XP, RP), 0.10, VEN = 0.00

C DRAW TICK MARKS FOR SCALE REFERENCE.

CALL PLOT (X1, Y1 + 2)

500 IV = IV + 2

C DRAW HORIZONTAL AXIS TITLE.

CALL SYMNL (01, 02, 0.20, 0.1, 20)

C DEFINE STAMP POINT OF FIRST SQUARE BAY.

ST = 0.1

CALL SYMNL (01, 02, 0.10, 0.0, 0.0)

ON 250 X = X + 0.20

C DRAW VERTICAL REFERENCE LINES FOR SQUARE BAYS.

CALL PLOT (X1, Y1 + 2)

550 Y1 = Y1 + 0.20

X1 = X1 + 0.20

C DRAW SECOND HORIZONTAL BAY SIZE FIGURES.

ST = 0.2

C DRAW A CYCLE FOR BOTH METHODS OF COST.

ON 700 I = I + 1

C DRAW A CYCLE FOR EACH TYPE OF BAY.

6 T0 100 C = 1,10

C DRAW THE GRAPHS OF FIRST SPANNING METHOD FOR A CYCLE.

CALL SYMNL (01, 02, 0.10, 11, (1), 0.0, 0.2)
C DRAW THE GRAPH OF SECOND SPANNING METHOD, FOR A CYCLE.
675 CALL SYMBOL (01) 03, 0.10, IFIG(2), 0, 0, 2)
690 CONTINUE
P2 = AC4(L)
J2 = R24(4)
C WITH PEN UP GO TO ACTUAL SQUARE 'RAY POINT.
C CALL PLOT (P2, P3, 3)
700 CONTINUE
J2 = J2 + 1
Q1 = R14(J2)
Q2 = R24(J2)
C DEFINE NEXT POINT OF SQUARE 'RAY.
C CALL SYMBOL (01) 02, 0.10, IFIG(1), 0, 0, 1)
C CALL SYMBOL (01) 02, 0.10, IFIG(2), 0, 0, 1)
M1 = M1 + 1
IF (M2 GT 1) GO TO 610
Q1 = 12.0
Q2 = 7.0
Q1 = 1.0
Q5 = 1.10
Q6 = 0.5
C DRAW GRAPH REFERENCE.
C CALL SYMBOL (01) 02, 0.20, TITL(1), 0, 0, 2)
C CALL SYMBOL (01) 02, 0.20, TITL(2), 0, 0, 2)
C CALL SYMBOL (01) 02, 0.20, TITL(3), 0, 0, 2)
C CALL SYMBOL (11) 02, 0.20, TITL(4), 0, 0, 2)
C CALL PLOT IN ORDER TO SET A NEW ORIGIN, AND 'FORCE BUFFER TO PRINT ALL.
C THE STUFF OF GRAPH TITLES.
C CALL PLOT (30, 0.0, 0.0, -3)
C RETURN
END
C SUNSHINE PLOT COPYRIGHT 1968 CALIF. COMPUTER PRODUCTS
C MODIFIED FOR HURDOUGHS B-5500 FORTRAN IV
C PLT 0098 9/28/68
C
C PLAT WINTINE FOR 760/500/2 PLOTTING SYSTEM
C 760 CONVERSION BY RUP GIESE, CALCOMP HOUSTON,
C PROGRAMMED BY KENNY STRANG
C
Bibliography.

FUNDAMENTALS OF OPERATIONS RESEARCH.
Russell L. Ackoff and Maurice W. Sasieni.
John Wiley & Sons. 1968

THE COMPUTER IN ARCHITECTURE.

COMPUTER RESEARCH SURVEY.
The American Institute of Architects. N.C. Research Group. 1968

PROGRESS IN OPERATIONS RESEARCH.
Julius S. Aronofsky (Editor). John Wiley & Sons. 1969

TIME SHARING SYSTEM. TERMINALS USER'S GUIDE. # 1038205.
Staff. Bourroughs Corporation. 1968-1969

INFORMATION PROCESSING SYSTEMS. FORTRAN COMPILER REF. MANUAL
# 1032083.
Staff. Bourroughs Corporation. 1968.

COMPUTERS IN ARCHITECTURAL DESIGN.

INTRODUCTION TO OPERATIONS RESEARCH.
C. West Churchman, Russell L. Ackoff and E. Leonard Arnoff.
Wiley International Editions. 1957.

INFORMATION NEEDS: OVERVIEW.

INFORMATION NEEDS: ROOM INVENTORY.
Staff. E. F. L., Duke University, and C.R.S. 1969

COMPUTER AIDED CAMPUS PLANNING FOR COLLEGES AND UNIVERSITIES.

COMPUTERS. INTRODUCTION TO COMPUTERS AND APPLIED COMPUTING
CONCEPTS.

THE SCIENTIFIC APPROACH.

COMPUTERS AND THEIR USES.
William H Desmonde. Prentice-Hall, Inc. 1964
COMPUTERS METHODS IN CIVIL ENGINEERING.
Steven J. Fenves. Prentice Hall, Inc. 1967

OPERATIONS RESEARCH AND SYSTEMS ENGINEERING.
Charles D. Flagle, William H. Huggins and Robert H. Roy

SYSTEM SIMULATION.
Geoffrey Gordon. Prentice Hall Series In Automatic Comp. 1969

DOCUMENTATION STANDARDS.

INTRODUCTION TO PROBABILITY AND STATISTICAL DECISION THEORY

COMPUTER APPLICATIONS IN ARCHITECTURE AND ENGINEERING.

THE COMPUTER AND THE ARCHITECTURAL PROFESSION.

STATISTICS.

INTRODUCTION TO OPERATIONS RESEARCH.
Frederik S. Hillier and Gerald J. Lieberman.
Holden-Day, Inc. 1968

ARCHITECTURAL PROGRAMMING. PROBLEM DEFINITION AND ITS RELATION TO DESIGN PROCESS.

PROGRAMMING THE IBM 1130 and 1800.
Robert K. Louden. Prentice-Hall. 1967

ABC'S OF COMPUTERS.
Allan Lytel. Howard W. Sams/The Bobbs-Merril Co. 1968

INTRODUCCION A UNA TEORIA DEL CONOCIMIENTO DE LA ARQUITECTURA Y DEL DISENO.

COMPUTER MODELING AND SIMULATION.

A GUIDE TO FORTRAN IV PROGRAMMING.
Daniel D. McCracken. John Wiley & Sons. 1965

FORTRAN WITH ENGINEERING APPLICATIONS.
SYSTEMS ANALYSIS. A COMPUTER APPROACH TO DECISION MODELS.

STRESS: A USER’S MANUAL. A PROBLEM ORIENTED LANGUAGE FOR STRUCTURAL ENGINEERING.
The Department of Civil Engineering. M.I.T. Press. 1964.

CPM IN CONSTRUCTION MANAGEMENT.

A FORTRAN PRIMER.
Elliot I. Organik. Addison-Wesley. 1966.

COMPUTER DRAFTING SPEEDS MOTEL DESIGN.

IN SEARCH OF URBAN EXPERTISE.

PARAMETERS FOR A COMPUTER PROGRAM.

PROBLEM SEEKING.
William M. Pena and John W. Focke. C.R.S. publication. 1969

A PLAIN MAN’S GUIDE TO SYSTEMATIC DESIGN METHODS.

COMPUTERS IN SCHOOLS OF ARCHITECTURE.

USING THE COMPUTER FOR DESIGN.

OPERATIONS RESEARCH; METHODS AND PROBLEMS.
Maurice Sasieni, Arthur Ysapan and Lawrence Friedman.
Wiley Toppan. 1959.

STATISTICS.

DIGITAL COMPUTER PROGRAMMING.

A MANAGEMENT GUIDE TO PERT/CPM.
Jerome D. Wiest and Ferdinand K Levy. Prentice-Hall Inc. 1969