INTERACTIVE COMPUTER MODELING OF THE BUILDING DESIGN DEVELOPMENT PROCESS

By

RICHARD EDWARD WEBB

Bachelor of Science in Architectural Studies
Oklahoma State University
Stillwater, Oklahoma
1977

Master of Science
Oklahoma State University
Stillwater, Oklahoma
1979

Submitted to the Faculty of the Graduate College
of the Oklahoma State University
in partial fulfillment of the requirements
for the Degree of
DOCTOR OF PHILOSOPHY
July, 1984

Thesis
1984D
W367i
cop.2

OKLAHOMA STATE UNIVERSITY LIBRARY

INTERACTIVE COMPUTER MODELING OF THE

BUILDING DESIGN DEVELOPMENT

PROCESS

Thesis Approved:

Philip M. Wolfe
Thesis Adviser

Kenneth E Case

Wayne Clemens

Carl B. Estes

Allan Brunk

Norman N. Durham
Dean of the Graduate College

1205550

© Copyright

by

Richard Edward Webb

July 1984

ACKNOWLEDGMENTS

I wish to express my appreciation and love to my wife, Sandra, and to both my daughters, Megan and Jaclyn, for their patience throughout my program. In addition, I would like to thank my wife for the hours spent typing this document.

I would like to thank each of my committee members for their guidance. A special thanks to my advisor, Dr. Phil Wolfe, for his help with the programming phase of this research, and to Mr. Alan Brunken for his insight and creativity.

Finally, I want to thank my parents for continued support, even if they did think I was crazy for pursuing this degree in the first place. At times, I agreed.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

The computer has been used by both the industrial engineer and the architect as an aid in the development of building designs since the early 1970's. Industrial engineers have created a number of computer aided layout techniques which generate building design concepts based primarily upon material flow or user defined departmental relationships. Where material flow is the dominant decision criterion these computer models have provided satisfactory results. However, in the design of offices, fire stations, restaurants, hospitals, shopping malls, and other building structures where material flow is not the major concern, the computer results have been poor and designers, therefore, have avoided using the existing techniques.

In response to these perceived model shortcomings, the architectural community created several computer aided design techniques which allow an increased number of decision criteria, including room proportion, visual access and department adjacency. However, in general architects are not using the computer as a design generator, but instead as a highly sophisticated drafting machine. A possible reason is that no computer technique has yet been introduced which models their building design development process.

This research has developed a computer assisted building design technique, combining the interests of both architects and industrial engineers, by modeling the building design techniques successfully

practiced by both groups. Selected design criteria and circulation patterns have been used to construct a design technique applicable to a wide variety of building types. Most importantly, this computer aided technique is programmed for a graphics computer system which allows the designer to see the design develop and encourages interaction during the development phase. It allows the designer to input design criteria directly affecting the building's function, quickly develop several design concepts and select the most appropriate. The absence of these features has been the primary obstacle to widespread use of computer aided design in many architectural offices. (6)

Statement of Need

Architects and industrial engineers share common ground in the area of space planning, as both are concerned with designing efficiently utilized space. However, because of background and training, each approaches the design problem differently. Industrial engineers use a variety of quantitative techniques to establish building layout concepts, while architects often rely more heavily upon qualitative measures. The computer aided building design models of both groups reflect their different approaches.

The number of computer techniques available is large and increases each year. A partial list is shown in Table I.

Many of these techniques have similar design algorithms. In general, the quantitative techniques are divided into two major classifications -- material flow and departmental relationships. For example, both CRAFT and COFAD use material handling costs as the primary design criterion. (19) Areas are rearranged within the building outline and

the corresponding material flow costs calculated. When no decrease in cost occurs, the program prints the resulting layout. These techniques, when applied to situations involving heavy material movement and when realistic handling cost data are available, yield good results. (21)

TABLE I

SELECTED COMPUTER AIDED DESIGN TECHNIQUES

ALDEP	LSP
COFAD	OFFICE
COMPROPLAN	PLANET
COMSBUL	PREP
CORELAP	RELATE
CRAFT	RMA Comp I
DOMINO	SPACE
IMAGE	SPACE CRAFT
LAYOUT	SOMI

The second major group of quantitative techniques arrange layouts based upon departmental relationships. The user defines and weighs the requirements for "closeness" between each department. The technique places departments into the layout, attempting to satisfy these relationships. ALDEP and CORELAP are common examples of this type. (19), (25)

The qualitative routines use design criteria similar to those used by architects when developing their design concepts. For example, GSP and IMAGE were developed by architects and utilize a wide variety of design criteria. GSP uses criteria such as adjacency, distance,

sight, access, and orientation. IMAGE criteria includes visual access, proportion, enclosure and circulation.

Both design philosophies have merit and have been used to solve selected design problems. However, three problems appear to be preventing general use of these routines. The first concerns design criteria. No technique combines the important decision principles of both industrial engineering and architectural routines. Therefore, shared usage of each group's techniques is limited, as illustrated by the current Institute of Industrial Engineering monograph entitled, Computer Aided Layout: A User's Guide, which contains only those routines of a quantitative background. Likewise, architectural designers have been quoted as saying they do not trust the output of engineering design routines which rely solely on material flow or user defined departmental relationships. (24)

The second problem is the interaction capabilities of many routines. Designers create and therefore, must be able to record their ideas quickly, accurately and as they are generated. The computer aided technique must allow the user to interact frequently, if desired, and instantly evaluate the impact of the designer's idea. Many of the established routines output printouts which must be analyzed, modified and then resubmitted to check for improvements. This long process, in effect, destroys the creative nature of design.

The third problem is possibly the most severe. Current computer aided techniques do not allow the designer to quickly order his design concepts or to test several concepts against one another. For example, after entering all the necessary design criteria, current techniques attempt to organize the design based upon a predetermined algorithm,

leaving the user very little choice. The user is not allowed to design.

Computer aided design techniques, coupled with computerized drafting programs can revolutionize the designer's job. They should decrease the design time and the time required to prepare construction drawings. Accuracy should increase as well as the quality of the final product. Computer techniques will not replace the designer, instead they should make his job more productive.

Research Goals and Objectives

A wide variety of computer aided design techniques have been developed, none of which has enjoyed widespread use. Teicholz, an architect and Associate Professor at Harvard University's Graduate School of Design, offered the following concerning the use of computer aided design in the architectural community. (34)

1. There is an unfounded fear that machines somehow threaten the creativity of architects.
2. It is difficult and probably impossible to computerize aesthetics.
3. The average design office has limited capital for computer models.

The primary goal of this research was to develop a computer assisted building design technique which addressed these constraints. The popularity of such a tool is dependent then upon three major features: the usability of the final computer-generated design, the ease of use by the designer, and the cost of the system.

A secondary goal was to integrate the design philosophies of industrial engineers and architects. As discussed previously, because of their training and background, each group approaches a design problem differently. Industrial engineers, typically, focus on design methods which utilize quantitative analysis while architects emphasize a more qualitative approach. Both methodologies result in answers to design problems, however, a technique which is founded on both qualitative and quantitative design decisions should produce improved building utilization, as well as increased acceptance among all designers.

Computer aided design techniques are not currently a replacement for designers -- they cannot create. They only attempt to satisfy user defined criteria. However, a technique which works alongside the designer, following his train of thought should make that designer's job more productive. Computers can help the user "think" through a problem and allow him to quickly analyze the merits of several alternatives. Already, sophisticated computerized drafting systems are allowing draftsmen to develop building plans as much as ten times faster, thereby decreasing their design fees 10 to 60 percent while increasing the profit margins at the same time. (6)

Therefore, the primary objective of this computer technique was to allow the designer to make timely interactions with the computer generated design. The technique will allow the user to modify the design during its development, while continuing to attempt to satisfy the design criteria initially entered.

"A picture is worth a thousand words" is particularly true in design. The designer, to be effective, must see the design develop.

Therefore, another objective of this research was to develop this design technique utilizing the features of a graphics computer.

A secondary objective was to develop a technique which may be used by the average design firm. Based on current literature, no computer aided layout technique has been written for the new small microcomputer systems. This technique is designed using the IBM Personal Computer but may be run on other personal computer models with minor program changes. Personal computer systems are available for less than $10,000 while larger more sophisticated systems may cost $30,000 to $100,000.

Summary of Results

To the knowledge of the writer, this research represents the first attempt to combine the strengths of architects and industrial engineers into the development of a computer assisted building design program. The technique is the beginning of a new concept of computer design, where the computer reacts like the designer and allows the user power to interact.

The current capabilities of the computer technique are :

1. The user may define the relationships between all activities located within the building structure.
2. The user may define the physical characteristics of each activity including area, proportion, ceiling height, floor level, wall assignments and any adjacency requirements.

3. The user may view each activity and modify its shape or scale.
4. The user may select from four circulation patterns: spine, courtyard, radial, and cluster.
5. The user may interact after each computer generated activity placement or allow the computer to create the design concept without intervention.
6. The user may store multiple designs on floppy disk storage for future reference or modification.
7. The user may generate a site plan including both easement and zoning setbacks, as well as, trees to be saved.
8. The user may locate and orient a selected floor-plan on the site. Both translation and rotation are allowed.
9. The user may display and modify elevation views of the generated building design. Activities can be ordered based on form and function.

These capabilities should make this computer assisted building technique an effective tool for building design concept development. It is the first step towards removing sketch paper from the designer's office.

CHAPTER II

LITERATURE REVIEW

Architects have shaped the environment we live in throughout recorded history. The Pyramids, the Parthenon, the Colosseum, and the great cathedrals of Europe serve as timeless examples of their skills. The early architect was designer, draftsman, engineer and building contractor. He was intimately involved in all phases of the design-construction process.

In vivid contrast, modern building design is the culmination of the efforts of architectural designers, structural, mechanical, and industrial engineers and a variety of support personnel including data processing, project management and payroll. Coordinated by the architect, they form a design team. No longer does the "master architect" instruct every detail and no longer does construction require years for completion.

These organizational changes are a direct result of today's complex building systems. Construction documents for commercial and industrial buildings can contain volumes of information, ranging from the size and type of door to be hung in a particular office, to specifications of steel beam sizes for the building skeleton. A typical set of these documents contains hundreds of drawings, including floor plans, building elevations, stair details and numerous sketches specifying construction procedures. A design

team requires months to develop this information, whereas, the "master architect" could take years.

Today's architectural clients cannot afford to wait years for the completion of their buildings. Time costs money -- money spent on architects, engineers and contractors, as well as, lost opportunity costs. It is these increasing costs that make computer aided design increasingly attractive.

A Philosophy of Design

Design is not a magical power possessed by a select few, but rather, it is a systematic problem solving process involving form and function. We all design. We rearrange living room furniture; we plan our route to work; we organize our desks and we landscape our yards. Design is defined as the act of putting things together into a meaningful relationship. (8)

A number of books have been published outlining approaches used to teach design. A widely accepted design approach is entitled, "Ordering". Ordering involves the arrangement of elements based upon the characteristics of those elements and upon established design criteria, guiding the creative process. For example, if applying the ordering process to arranging books on a desk, the elements would be the books; a characteristic might be a measure of the frequency of use; and the design criteria might be to put the most used books in front and the least used at the back. The selection of different element characteristics and design criteria will result in different designs.

For building design, the elements, characteristics and design criteria can be divided into four ordering systems: function, space,

circulation, and context. The function system involves the arrangement of elements, or building activities, based upon the desired relationships of activities with each other. These departmental relationships may depend upon one or a combination of several element characteristics ranging from supervisory requirements to the volume of pedestrian or material flow.

The space ordering system arranges activities based upon their volumetric characteristics. These characteristics include the activity's area, the shape of interior space and the scale. The designer then groups activities into the same space due to strong relationships, or zones activities into separate spaces due to conflicts.

The third ordering system is circulation. Spaces may be geometrically ordered using either linear or point systems. Linear systems arrange activities along and in relation to a line. Activities can be arranged on both sides or just one side of the line. The line can be straight, curved or offset. The point system arranges spaces around a nucleus. The arrangement of spaces using either system is based on a combination of the characteristics of each space. Characteristics may include number of edges, heights, widths, area, centroids or shape.

Context is defined as the whole situation, background or environment relevant to some event or product. (8) The context ordering process arranges the building's activities based upon the characteristics of the building site. Characteristics may range from utility setbacks to access to the sun or to a desired view.

Background of Computer Aided Design

Armour and Buffa introduced CRAFT (3) (Computerized Relative Allocation of Facilities Technique) in 1963, the first published computer aided layout routine. Architects were independently paralleling the industrial engineering efforts, as Sutherland introduced the first graphic sketch pad in 1964. (9)

Current computer aided design techniques are idea generators, as they often help the designer develop his initial layout concepts. As shown in Figure 1, the output must be modified into a feasible building form. Computer aided layout techniques are primarily used during the initial design stage to determine the basic departmental arrangements within the building complex.

As briefly discussed in the previous section, computer techniques can be classified into two general categories -- construction and improvement. An improvement routine evaluates an existing layout, modifies the design and then reevaluates the changes. COFAD, CRAFT, OFFICE, and SPACE CRAFT are examples of improvement techniques. In contrast, a construction technique does not rely on initial departmental arrangements. Construction techniques are primarily used for new construction, however, many can be used in remodel design by predefining the location of selected departments which are not to be included in the proposed modification. Most of the existing computer routines are in this category, including ALDEP, CORELAP, GSP, IMAGE, PLANET, RMA Comp 1 and SOMI. (31)

Lee, further classified computer aided design techniques. (20) The first, entitled the "Interchange Technique", used an existing layout and systematically conducts pairwise or three way exchanges and reevaluates

NOTE: Lines are not constructed by the routine.

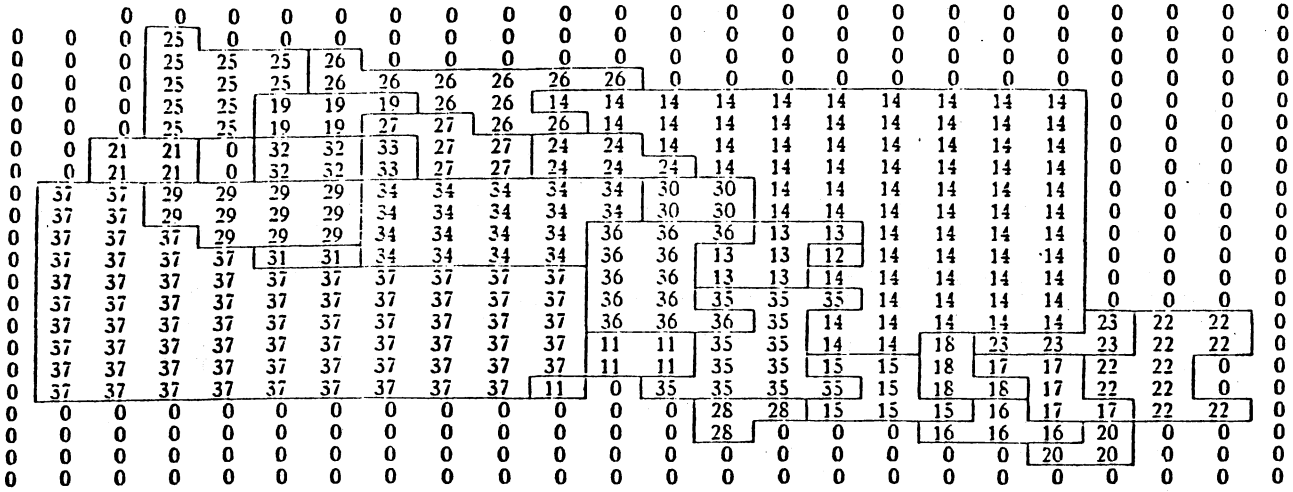


Figure 1. Typical Final Layout Design

the design based upon a selected criteria. CRAFT, COFAD and SPACE CRAFT are grouped into this category.

The second type, labeled "Neighboring Technique" selectively places departments around a centrally located area, based upon the entering department's relationships with previously placed areas. Examples include CORELAP, COMSBUL and RELATE.

The third category is best examplified by ALDEP. The "Random Technique" operates much like the Neighboring Technique, except the initially assigned department is selected at random, rather than based on relationship scoring.

The "Vector Technique" is the fourth category. This technique considers relative distances between departments to generate a space layout. Distances are expressed in matrix form and the program places the department with the lowest total distance in relation to all others. The routine then finds and places departments that have the shortest distances to the initially placed department. COMPROPLAN is the most advanced routine in this category.

The final classification is the "Multi-Constrainted Technique", which are routines that use several criteria for department placement. These routines generate space layouts by modifying the designs to reduce violations in design criteria constraints. GSP (General Space Planner) and IMAGE are both grouped into this type.

Review of Selected Technique Methodologies

This section will briefly outline a technique from each of the previously described categories. A brief outline of all techniques listed in Table I is presented in Computer Aided Architectural Design.(24)

Interchange Technique - CRAFT

CRAFT, introduced by Armour and Buffa in the July 1963 issue of Management Science, is the first computerized layout technique. CRAFT is a hueristic procedure that used material handling costs as the major design criteria. The input requirements of CRAFT include (1) material flow, in terms of a selected unit per time period, between each area of the building, (2) the cost of material handling per load moved per unit distance, and (3) an initial layout indicating existing department orientations and space requirements.

After reading the input data, the CRAFT routine calculates the movement and movement costs between centroids of all departments, determining the total cost of the initial layout. Next, CRAFT performs either pairwise or three way exchanges of existing departments and recalculates the total material handling cost. If the total cost is reduced, the layout is reconstructed and the exchange process begins again. When no further cost reduction resulting from the exchange process is found, the routine terminates and the final layout is printed. The routine is path-oriented and therefore, does not consider all possible exchanges and may yield a sub-optimum result. This however, is not a major problem since the CRAFT output must be modified before construction can begin.

Numerous programs have used CRAFT as a foundation. The strength of CRAFT is its final layout because it suggests a design to minimize material handling costs, which are major no-value added costs. The major disadvantages include (1) the need for an initial layout, (2) material handling cost estimates (difficult, at best, to obtain),

(3) a single design criteria and (4) an irregularly shaped final layout.

Neighboring Technique - CORELAP

CORELAP (Computerized Relationship Layout Planning) was initially developed by Lee as a Master of Science Thesis at Northeastern University in 1966. (21) CORELAP is based on departmental relationships, utilizing the "REL Chart" developed by Richard Muther, as part of the Systematic Layout Planning (SLP) Technique. (26)

Inputs include (1) space requirements for each department, (2) maximum building length-to-width ratio and (3) a departmental relationship matrix. The relationship matrix is shown in Figure 2. As indicated, vowels are used to indicate the user defined relationship between each area. Muther (26) suggests the REL Chart be completed by interviewing personnel working in each area, determining relative material flows, and by identifying special considerations such as fire hazards, noise or privacy.

CORELAP layouts "grow" like crystals, outward from a central point. CORELAP begins by calculating which department has the highest relationship with all other departments, based upon the relationship matrix and values assigned to each vowel code. (Total Closeness Rating - TCR) The department having the largest TCR is then placed in the middle of the layout. The other departments are then searched and an area with an "A-relationship" is placed next to the initial department. The procedure continues to search for other departments with A-relationships between themselves and the initial department and places them into the layout. If no more A-relationships are located,

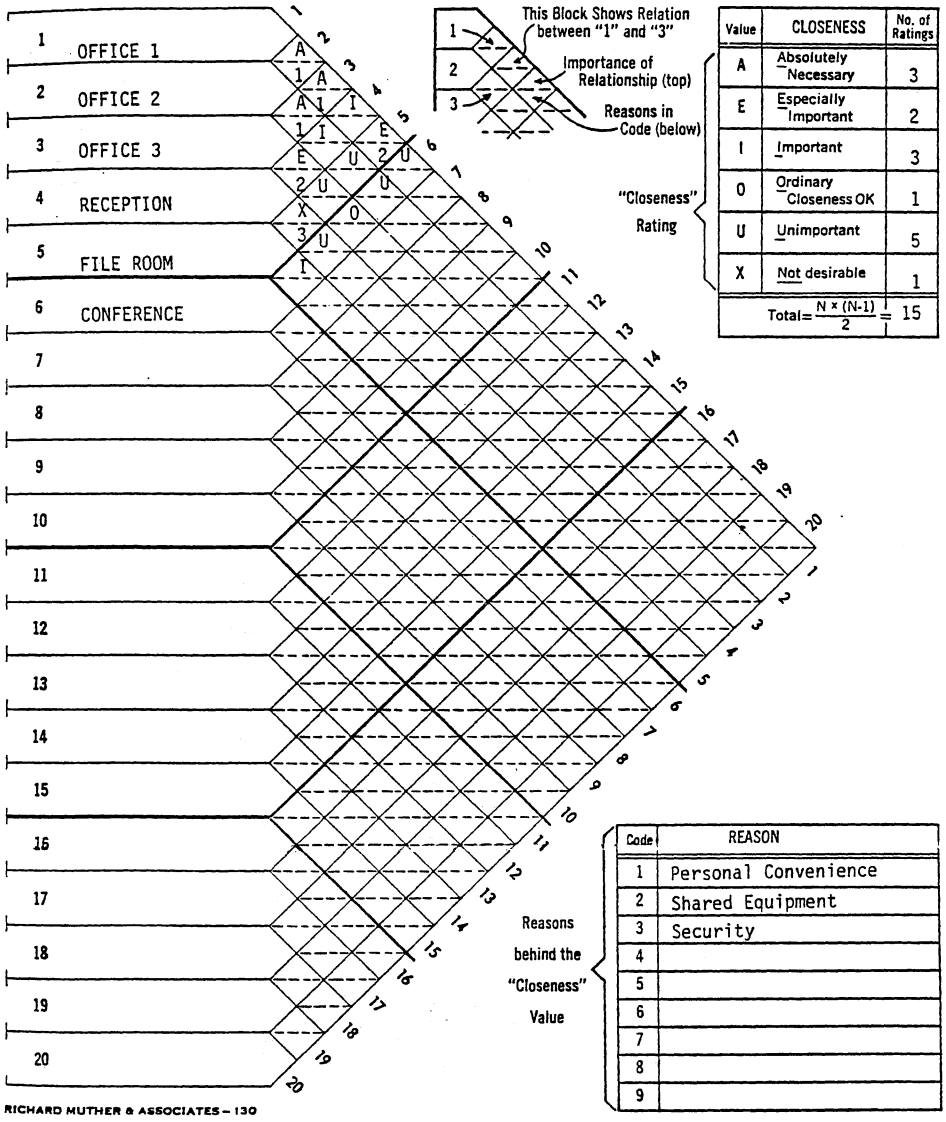


Figure 2. Sample Relationship Chart

departments not yet placed are checked against those departments previously placed. If no A-relationships are found at this stage, the routine begins back with the initial layout and checks for E-relationships. The procedure continues through the E, I, O, U and X-relationships.

As departments are selected, they must be placed into the layout. CORELAP uses a procedure based on the sum of the weighted relationships between the entering department and existing areas. (Weight values for this portion of the routine are user-defined.) All possible entry locations are checked and the entering department placed at the position with the largest entering value. Ties are broken using boundary length.

CORELAP uses a shortest path distance measure between departments as the evaluation criteria. The score of the final layout is the sum of the product of the shortest distance measures between departments and the corresponding relationship value.

The major advantage of CORELAP is the use of the relationship matrix. If critical information for a particular design is used, such as privacy required, information flow or personal contact needed, the relationship matrix contains broader information necessary at this stage of the design process. The major disadvantages are (1) irregularly shaped layouts, (2) limited design criteria, (3) aisles, elevators and stairwells are not included, and (4) the circulation pattern is not defined.

Random Technique - ALDEP

ALDEP (Authomated Layout Design Program) was developed by IBM in Rochester, Minnesota and first published by Seehof and Evans in the December 1967 issue of Industrial Engineering. ALDEP does not attempt to develop an optimum layout but instead produces many layouts, evaluates each and allows the user to select the best design.

The major input requirements of ALDEP are similar to those of CORELAP, as both operate using the relationship matrix. In addition, the user must input the building outline specification and the square footage requirements of each department. As with the CRAFT routine, selected departments may be preassigned to a specific location within the building. Lastly, the user must indicate the scale and the sweep width. (29)

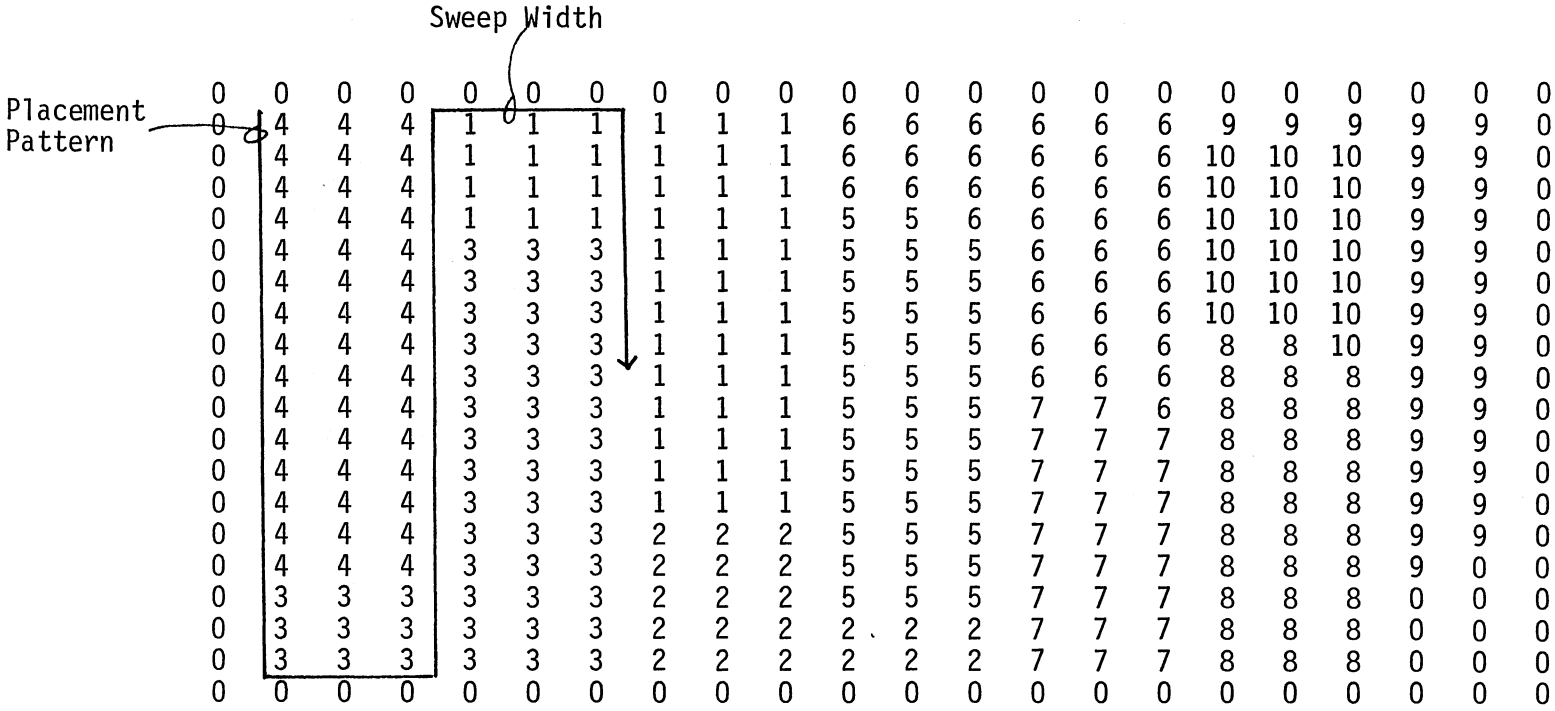
The ALDEP routine randomly selects a department from the list of those available for placement and places it in the upper left hand corner of the building outline. Depending upon the specified scale and sweep width, the program then begins to fill the layout as shown in Figure 3. Next, the relationship matrix is searched for a department having a relationship measure equal to or greater than the minimum value designated by the user. If a department is found, it is placed into the layout. However, if no department is found which meets the relationship requirement, another department is randomly selected and the process repeated.

ALDEP evaluates each layout based upon adjacency and relationship measures. That is, for each adjacent department, the relationship measure is determined (A, E, I, O, U or X) and a numerical weight

TRIAL LAYOUT 14 SCORE = 528

0 0
0 0

TOP FLOOR



GROUND FLOOR

0 0
0 0

TERRACE LEVEL

Figure 3. Typical ALDEP Layout Design

assigned (64, 16, 4, 1, 9, - 1024). The weights are then summed and the score assigned to that layout.

The major advantage of ALDEP is its ability to generate many different scored layouts on each run. The major disadvantages of the routine include (1) sweep method does not guarantee the relationship measures are obtained, (2) irregularly shaped departments (3) scoring method may not accurately assess the layout design, and (4) the circulation pattern is not defined.

Vector Technique - COMPROPLAN

COMPROPLAN was developed in 1971 by Perry, Dean and Stewart, an architectural firm based in Boston. This routine, like CORELAP and ALDEP, centers around the relationship matrix. However, COMPROPLAN specifies relationships by distances between departments. That is, a "short distance" is equivalent to an "A-relationship". COMPROPLAN is also interactive, allowing the user to exercise commands on the finished layout, such as "Move", "Rotate", "Implode" and "Explode".

COMPROPLAN selects the departments having the smallest total distance requirements between all other departments and places it at the origin of a three dimensional coordinate system. (At this stage of the routine, departments have no area and are represented by points.) The relationship matrix is then searched for the department having the shortest distance requirement with the initial department. This department is placed on the X-axis. The next department chosen has the shortest distance requirements between both the first and second departments. This procedure continues until all departments are located. Unlike CORELAP, COMPROPLAN arranges incoming departments in three dimensions when necessary.

After all departments are placed, the layout is projected onto the X-Y plane. Area, either circles or rectangles is added to each department. Alternatives are then developed by the user while in an interactive mode.

A strength of COMPROPLAN is its interactive capability, allowing the user to quickly and easily modify the design. The major disadvantage is that the solution is based on points and distances in three dimensions rather than two dimensions.

Multi-Constrained Technique - IMAGE

IMAGE was developed at the MIT Department of Architecture and was originally conceived by Johnson. The research was supported by the National Science Foundation from 1968 to 1974. The project employed from three to five people each year. (38) IMAGE is a routine which allows the user to input a wide variety of design criteria which is then used to determine the layout of the building.

The user of the IMAGE system must complete two input phases before running the program. First, each space must be defined by calling up a default square and assigning a desired scale and proportion. Next, the spaces are interactively moved into an arbitrary arrangement.

The user must next define the relationships and constraints between these spaces. He simply types in the desired design criteria for each space. Examples of this input include:

- "RELATE LOBBY VIA AREA ZOO"
- "RELATE BEDROOM BATH VIA ADJACENCY"
- "R KITCHEN DINING VIA SHAREDWALL"

The following are selected design criteria available with IMAGE. Individual criteria can be weighted in the event there are multiple criteria which cannot be satisfied for a particular space.

1. NEAR, FAR is used to control how far apart two spaces should be. The user inputs a maximum and a minimum desired distance. Exact distance placements are made by entering the same value for both maximum and minimum.
2. SHAREDWALL is used to partially align two spaces. It does not place spaces next to one another, but does line up departments on a common axis.
3. AREA dictates the area of the department. Both absolute measures can be used or one area can be specified relative to another. AREA preserves the desired proportion.
4. ADJACENT is specified if two spaces are to touch. Corner contact satisfies this criteria. Therefore, to ensure wall contact, ADJACENT must be used with SHAREDWALL.
5. ONTOPOF is used if two spaces are to be one on top of the other with no overhang.
6. FLOOR is specified if two spaces are to have floors at the same elevation.
7. KEEPOUT is input if two spaces cannot overlap.
8. OVERLAP is specified to allow, but not require, overlapping between two departments.
9. VISUAL ACCESS is used if two spaces must "see" one another. Windows can be modeled using VISUAL ACCESS by specifying

small window elements and relating them to selected areas using VISUAL ACCESS.

Many of these design criteria allow the user to input both minimum and maximum values. Finally, IMAGE will alter the original shape of spaces to satisfy a criteria constraint.

IMAGE creates layouts by modifying the arrangement of spaces in order to reduce the violations of design criteria constraints. IMAGE begins with a space and attempts to satisfy all criteria associated with it. Doing so, the routine attempts to arrange the next space and continues throughout all designated departments. If while operating on a space, a previously satisfied constraint must be violated to satisfy a present criteria, IMAGE determines a compromise using an optimization technique process called Least Mean Squares Fit. It uses the desired changes for each area and the appropriate weights of these changes to calculate new arrangements. After all spaces have been addressed, the final layout is printed and the user may then interactively rearrange the layout, modify the design criteria and resubmit the problem.

IMAGE evaluates a design based upon how each individual space satisfies its particular design criteria. No numerical score is calculated, however, the user has two routines, RANK and TEST, to aid in the design evaluation. RANK produces a list of spaces exhibiting worst or best conditions. For example, "RANK 10 WORST SPACES" could be entered by the user. Referencing the list then generated by IMAGE the user could then type TEST XXX, and the routine will outline the status of design constraints associated with the space XXX.

The strength of IMAGE is (1) its ability to involve multiple decision criteria in a design, (2) its interactive capability and

(3) its relatively straight forward operation. The major constraint of the system is that effective use depends heavily upon the design skill of the user and his knowledge of the design problem.

Requirements of Computer Aided Technique

A nationwide survey of designers was conducted to determine the features required by a computer aided design system. The following table, Table II, contains, in order of priority, the results of that study. (24)

TABLE II

COMPUTER AIDED DESIGN SYSTEM REQUIREMENTS

1.	User-oriented. It must be pencil-oriented and the user must be involved. There should not be a middle man. When a decision is to be made between "easier to program" and "easier to use", the latter is opted. The system should maintain the "learn by doing" concept.
2.	Interactive. The system should be used in a conversational mode and as much as possible it should be in a multiple choice style.
3.	Generic/flexible/versatile. The packages of computer programs must not only solve one specific problem, but many similarly related problems. It must be able to handle different building types.
4.	Convenient. Learning and training to use the system should not require an excessive amount of time.
5.	Efficient. The system should make good use of the hardware and software capabilities.
6.	Traditional. The system should not demand that the architect/planner deviate from the institution-dependent design process.
7.	Responsive. The input-output should require minimum time. As much as possible, each user action must be responded by the computer within seconds.
8.	Open-ended. The packages of programs should be designed as flexible modular building blocks. Deletion, addition, and reorganization to the software as well as hardware must not be a handicap.
9.	Economical. The cost of the complete system should be within the financial capability of the large architectural firms, whereas the various smaller versions of the system are for the smaller firms.
10.	Common data-base. As much as possible data should not be duplicated for each program package. Common data should be stored in a data base for the retrieval by those programs that need them.
11.	User control. The computer should not design. It should help the user to make the best design decision possible. The user is in command.
12.	Graphical. The architect/planner thinks in terms of graphics much more than in a textual or numerical fashion. The system must cope with this requirement.

CHAPTER III

PROGRAM OVERVIEW

Computers cannot design. They can, however, generate design concepts based upon space, relationship, circulation and form input requirements. As previously discussed, existing computer aided layout techniques offer extremely limited design flexibility. Often, the results of these techniques must be modified to an extent that using the computer is counter productive. In addition, most existing computer techniques are modelled around one placement routine. Therefore, a CORELAP layout always looks like a CORELAP layout, an ALDEP layout always looks like an ALDEP layout and so on. The result of this research, CREATE, is a computer aided building design technique which incorporates activity space requirements, desired relationships between departments, circulation needs, building form and orientation into a design decision process.

Technique Methodology

CREATE allows the user to develop building designs around four major circulation patterns -- spine, courtyard, radial and cluster. Each circulation pattern then results in a different ordering scheme and therefore, an alternative design concept. This flexibility allows the designer to evaluate several alternatives without resubmitting the program for processing.

CREATE is an interactive routine. The user can at any time interrupt the computer's design routine and make changes in the evolving concept. The computer then continues to develop the layout based on the input information. Allowing the user to interact during the placement routine increases the useability of the design and results in multiple concept alternatives within a selected circulation ordering pattern.

CREATE is a graphical computer aided building design technique. It represents building areas with line drawings, not clustered numbers. The layout is drawn on the computer screen as the computer generates it, allowing the user to see the design develop. Once the initial building plan has been developed it may be placed on a building site specified by the user. Trees and zoning setbacks are graphically illustrated. Finally, elevation drawings of the building plan may be viewed, and the plan modified into a more desirable form.

CREATE is programmed for the IBM Personal Computer. The initial cost of personal computers has made them extremely popular with American businesses. The computer used for this research has a color monitor, 256K of memory (only 64K required), a graphics card, a light pen and an EPSON MX-100 printer. Total cost of the system ranges between $4,500 and $5,500. In addition to being less expensive, personal computers have fewer job control and data management commands than larger main frame computer systems. Finally, CREATE is written in BASIC.

Program Structure

CREATE is structured into twelve major subroutines, as illustrated in Figure 4. A computer listing of the program is contained in Appendix A. The user initially has three options: (1) create a new building layout plan, (2) create a new building site plan, or (3) modify building site plan.

To create a new building plan, CREATE first calls the subroutine FUNCTION and the user inputs information about the relationships between building areas, or activities. The computer then determines a relationship score and displays the results for verification by the user. CREATE next calls the SPACE subroutine.

In SPACE, the designer inputs the physical characteristics of each activity. A plan and two elevation views for each activity are then displayed and the user prompted for any modifications. The designer can adjust the shape or scale of each activity. When complete, CREATE loads the CIRCULATION routine.

CIRCULATION contains the menu board for the circulation pattern selection. Tke designer selects the appropriate pattern -- SPINE, COURTYARD, RADIAL or CLUSTER -- and CREATE enters that design development routine.

Each of the design development routines contains a set of rules and guidelines for activity placement. CREATE attempts to follow these guidelines and prompts the user when unable to solve the problem within the constraints. The user may elect to have CREATE develop the building layout independently or he may select an option that allows changes to be made after each activity placement. When the design is

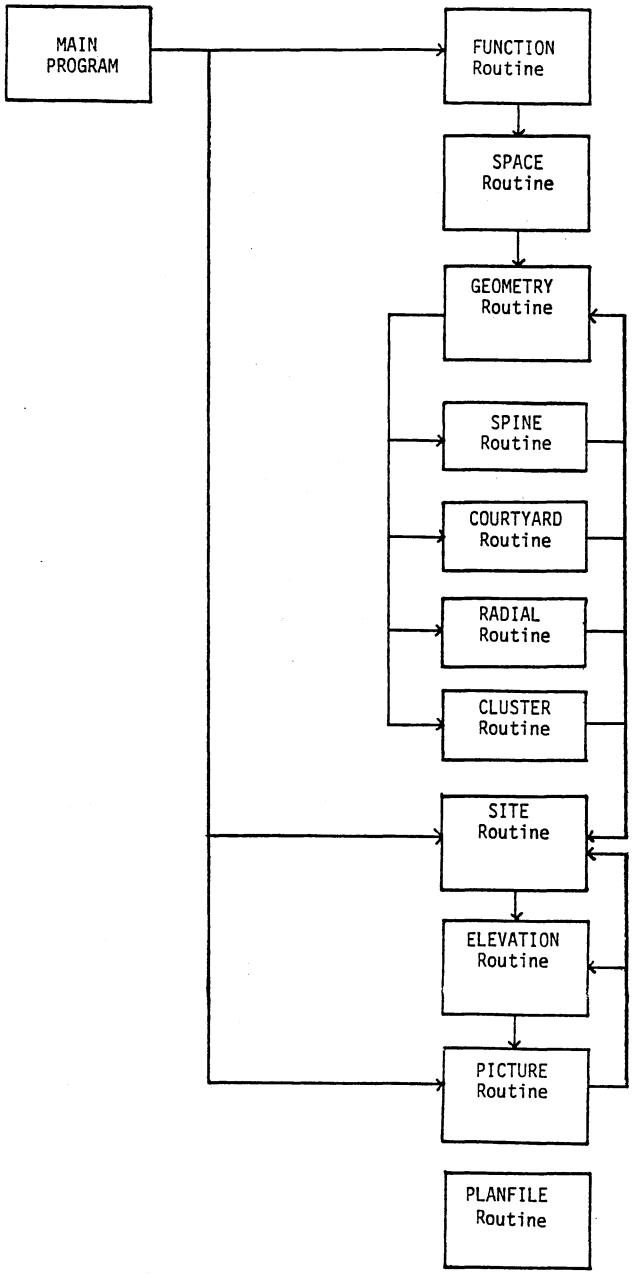


Figure 4. CREATE Program Structure

complete, the designer may store it for future reference and develop another layout design or he may place the activity on a building site plan.

Using the SITE subroutine, the designer inputs the building site coordinates, zoning setback points and tree locations. CREATE then asks the user to select which building plan to place on the site. Once placed, the designer may translate and rotate the building plan until he achieves the proper orientation. Once oriented, CREATE asks if elevations are desired.

The designer inputs the desired elevation view and indicates which activities are to be included and the ELEVATION subroutine draws the view. CREATE then allows the designer to translate activities based upon form considerations. These changes are stored and the PICTURE routine loaded.

The PICTURE routine is the modification portion of the CREATE program. The building plan view is displayed, as modified with the ELEVATION routine. The designer may then translate and rotate individual activities or the entire building plan. Satisfied with the changes, the designer then calls ELEVATION and repeats the modification process. Lastly, the user may elect to start over completely or just begin design work on another concept.

CHAPTER IV

DESCRIPTION OF PROGRAM ROUTINES

Reproductions of screen displays are used to illustrate user interaction and flow diagrams indicate the logic and information flow.

MAIN Program

The MAIN program has three basic functions. First, selected variables are defined and a brief description of the program is listed. Second, all major arrays are dimensioned, and third, the initial menu board is displayed.

As shown in Figure 5, the user has three options at the beginning of the program: Create Building, Create Plan or Modify Plan. As previously discussed, option one allows the designer to input relationships and physical characteristic data and develop a new building layout. Option two allows the user to develop a new building site plan and orient a previously stored building plan on it. The third option lets the designer study and modify a building plan and site plan already saved on disk.

FUNCTION Routine

The primary purpose of the FUNCTION routine is to develop the desired relationship rating between all pairs of activities to be included in the building plan. Figure 6 contains the FUNCTION flow

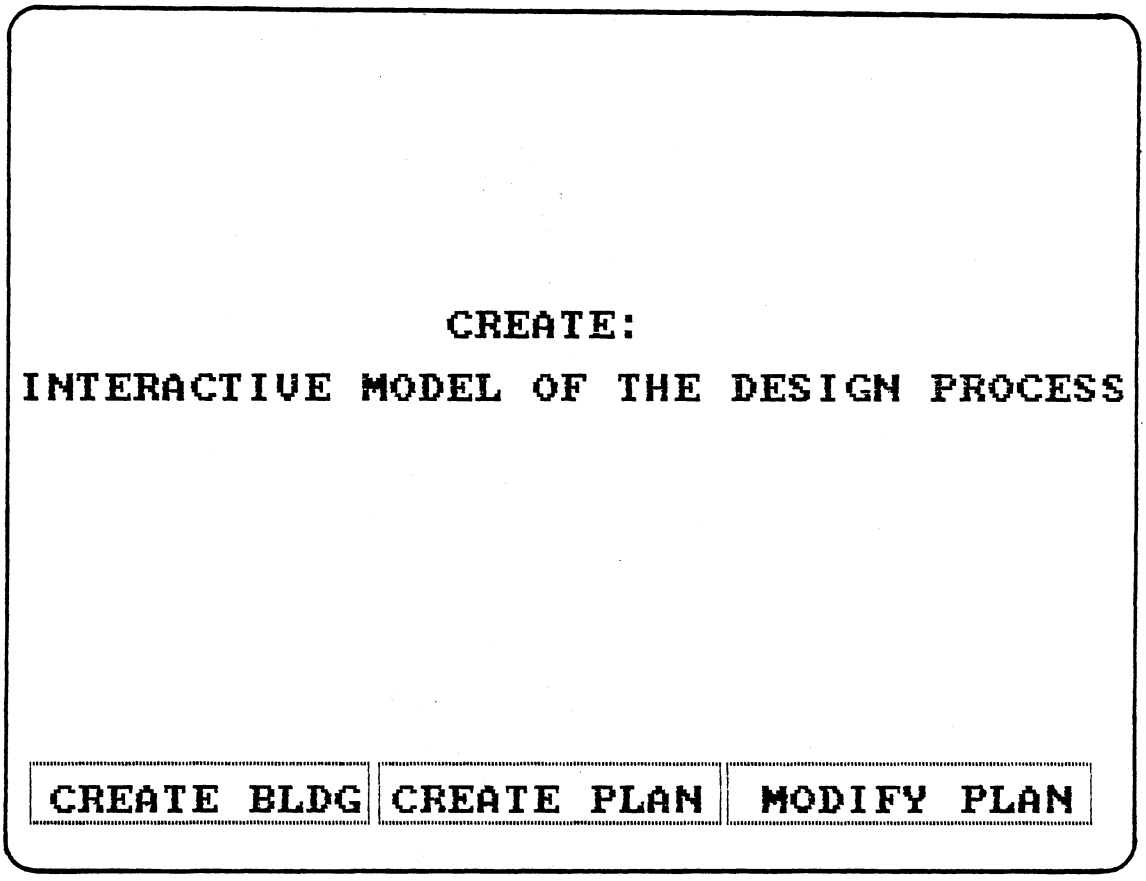


Figure 5. Main Program Menu Board

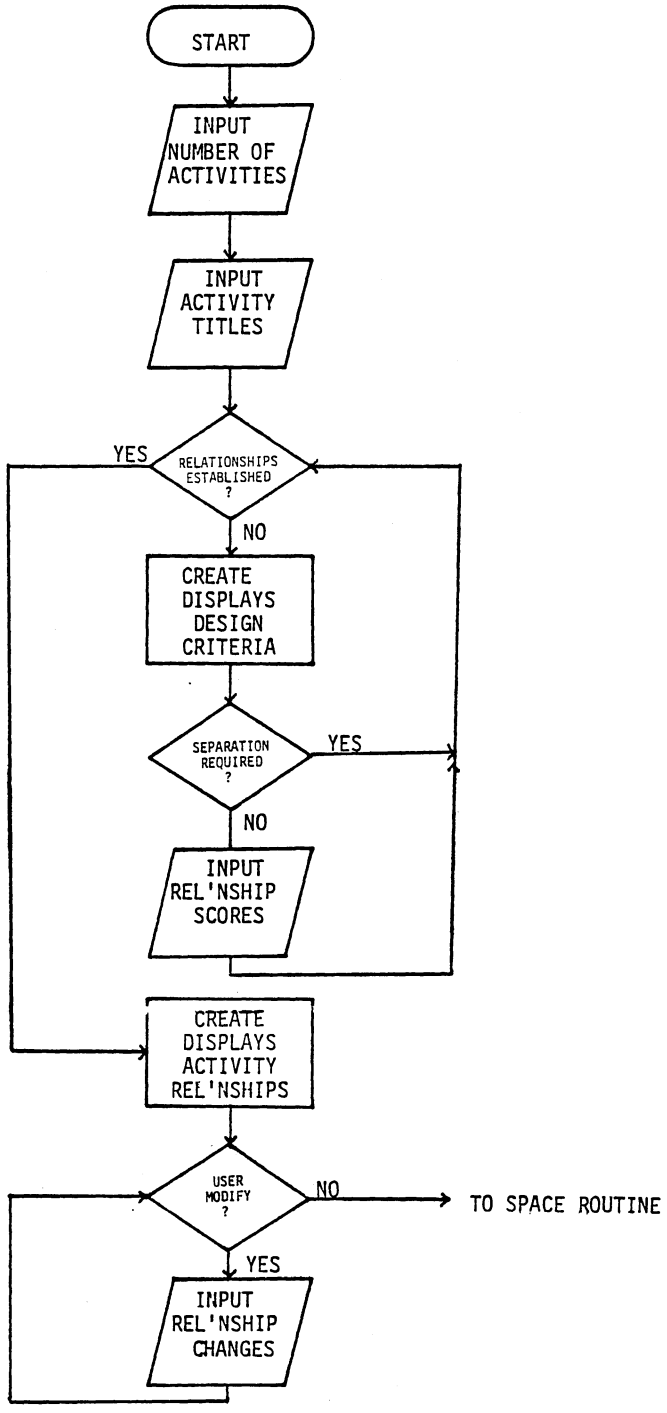


Figure 6. FUNCTION Routine Flow Diagram

diagram and Figure 7 contains the initial screen display of the FUNCTION routine.

CREATE first cycles through all pairs of activities, while the user establishes the desired relationship. Using the light pen, the designer indicates whether or not the two activities should be separated. If separation is not required, he then indicates the degree of importance -- None, Low, Medium or High -- for each of the nine displayed design criteria. For example, using Figure 7, if one secretary was assigned to the occupants of Office 1 and Office 2, the designer might rate the "Shared Personnel" design criteria as "High". Conversely, if each office has its own equipment except, perhaps, for a copier, the designer might give the "Shared Equipment" criteria a "Low" rating. CREATE cycles through the criteria allowing the user to rate each. CREATE then sums the indicated ratings and calculates a numerical score which will be used later to determine which activities should be adjacent.

Many of the early computer aided layouts represented relationship ratings using an eight character format shown in Table III. (26) Because of the number of relationship ratings, it is difficult for some users to distinguish between the rating levels. In addition, the author believes the "XX" relationship is ambiguous. To improve this format, CREATE uses only four assignments, also displayed in Table III.

Based on the numerical score, CREATE next assigns the appropriate alphabetic score and displays the results as shown in Figure 8. The user may modify these relationships by simply entering the pair of activities requiring change and the desired alphabetic relationship.

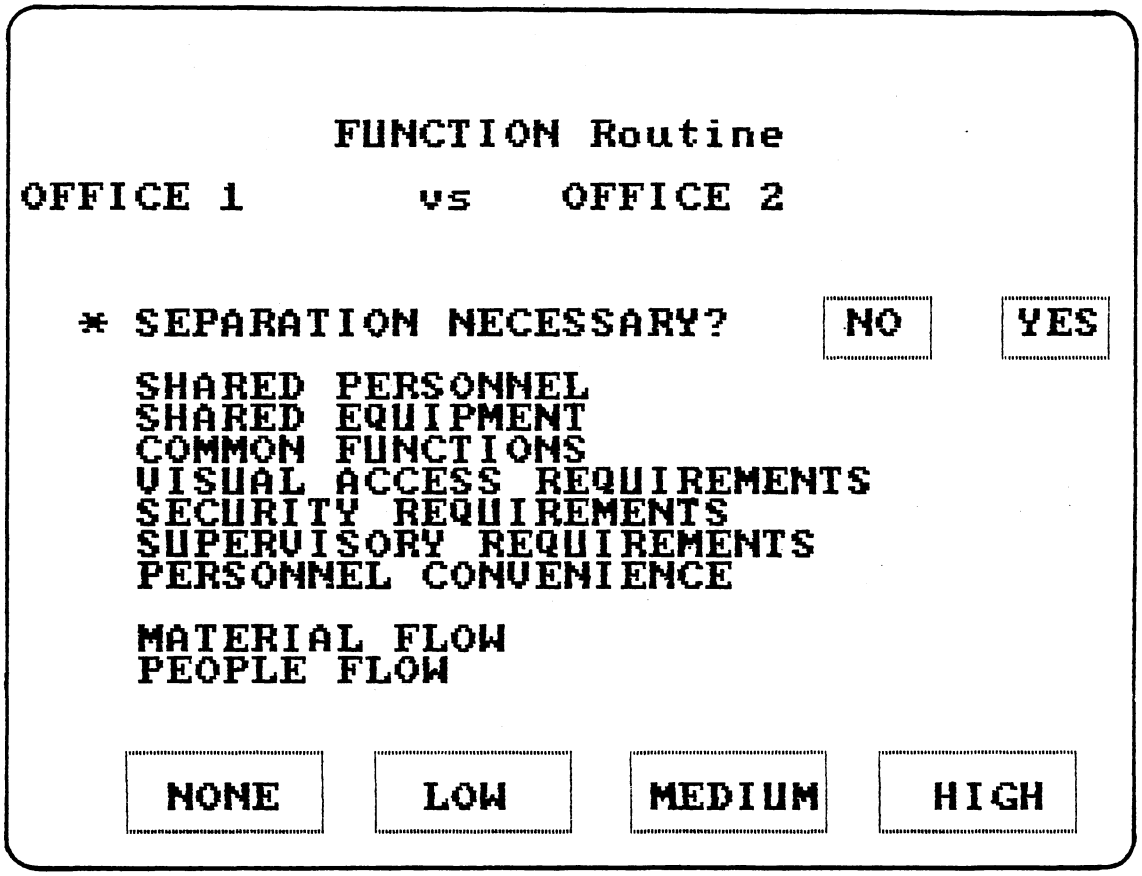


Figure 7. FUNCTION Routine Relationship Screen

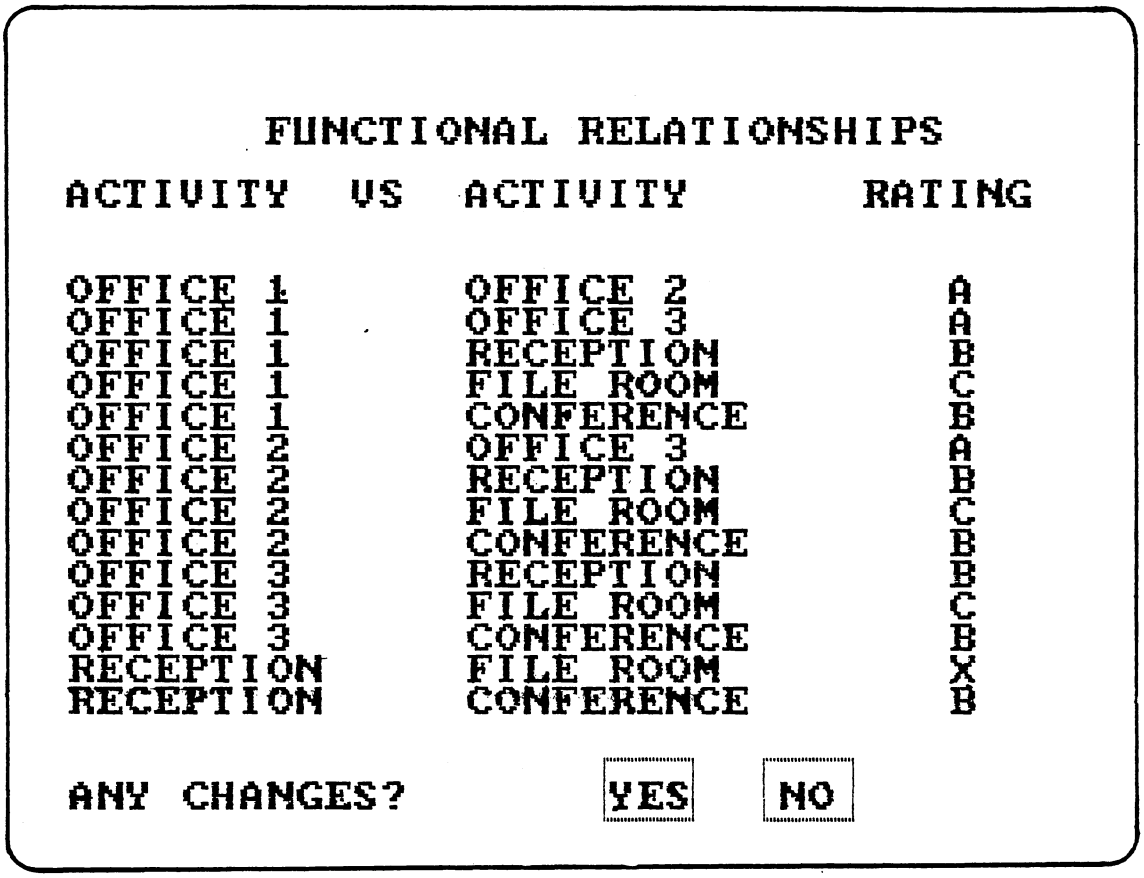


Figure 8. FUNCTION Routine Relationship Summary Screen

CREATE simultaneously changes the activities' numerical relationship score. After the relationships between all activities have been developed, CREATE calls the SPACE routine.

TABLE III

RELATIONSHIP RATING FORMATS

Relationship Routines Other	CREATE	Description
A		Absolutely necessary that the two activities be adjacent to one another
E		Especially important that the two activities be close to each other
I		Important that the activities be close to each other
O		Ordinary closeness only required between two activities
U		Unimportant that the activities be close to one another
X		Separation necessary between the two activities
XX		Separation required between the two
	A	Very important that the two activities be as close as possible to each other
	B	Important that the two activities occupy the same building vicinity
	C	Unimportant that the two activities be placed near each other
	X	Separation desired between the two activities

SPACE Routine

In the SPACE routine, the user inputs information describing the physical characteristics of each activity. Table IV describes the physical characteristic data inputs of the SPACE routine. The screen display is shown in Figure 9.

TABLE IV

SUMMARY OF PHYSICAL CHARACTERISTICS

Characteristic	Default Value	Description
Area	--	Square footage required by the activity
Proportion	1 to 1	Length to width proportion of the activity
Ceiling Height	10	Ceiling height of the activity
Floor Level	1	Indicates building story (currently single story buildings only)
Wall Assignment	--	Indicates desired building side for activity placement. If no assignment, CREATE selects wall location
Adjacency Requirement	--	Increases relationship score to insure adjacency achieved

The SPACE routine flow diagram is shown in Figure 10. CREATE cycles through each of the building's activities and asks the user to input the physical characteristic data shown in Table IV. As indicated, where appropriate, default values are entered when the user wishes to utilize them.

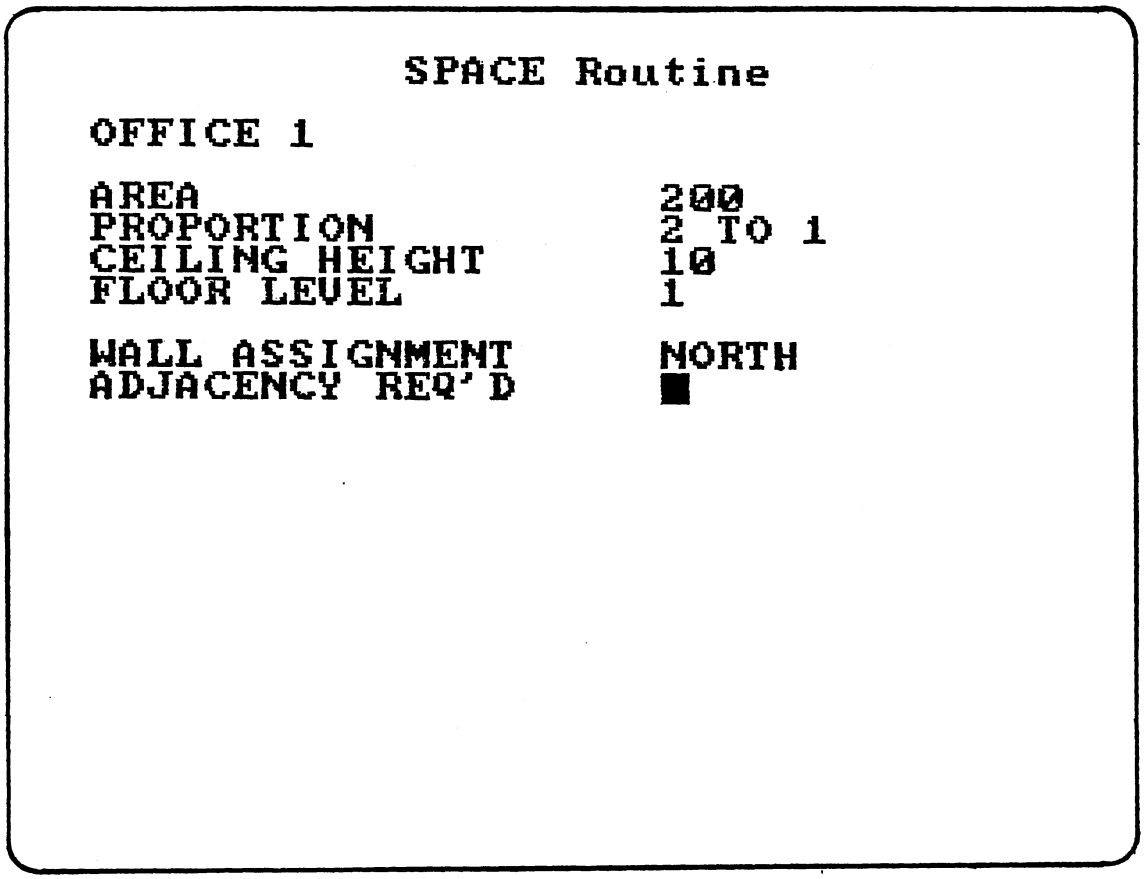


Figure 9. SPACE Routine Data Input Screen

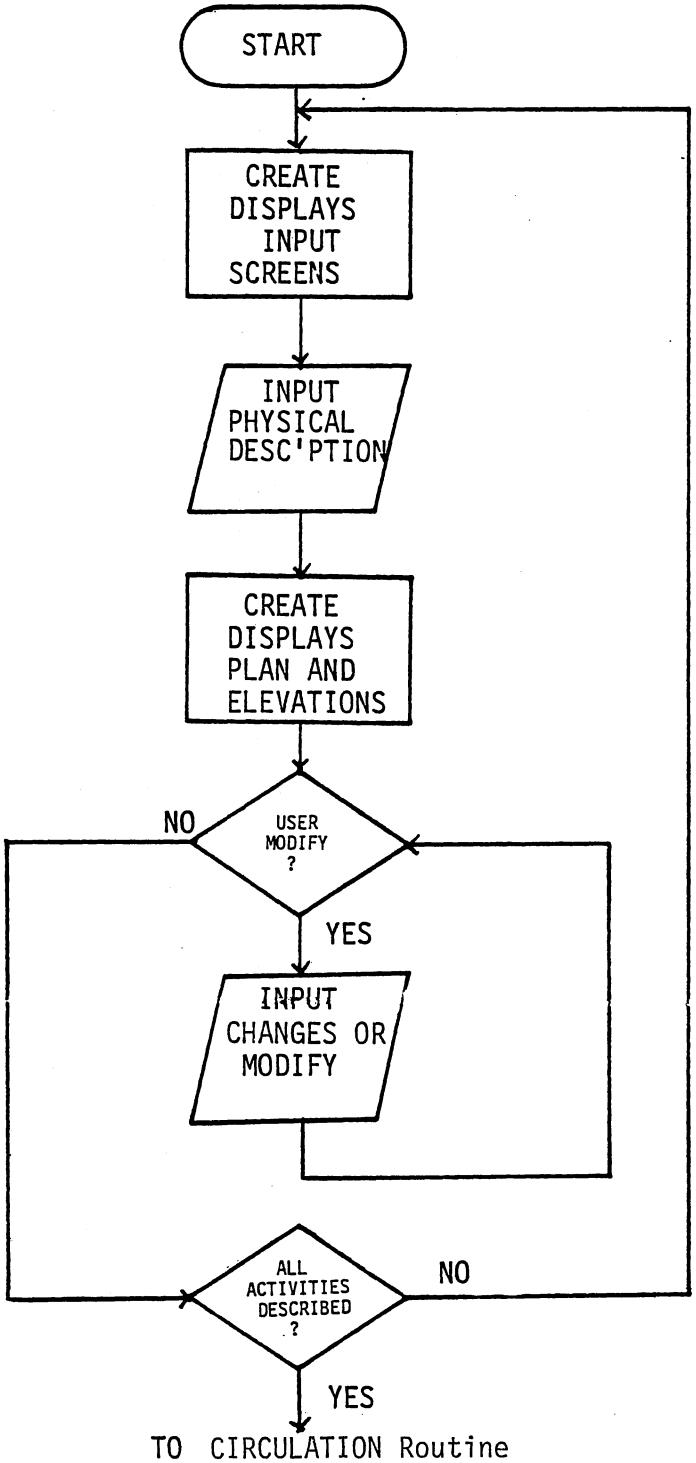


Figure 10. SPACE Routine Flow Diagram

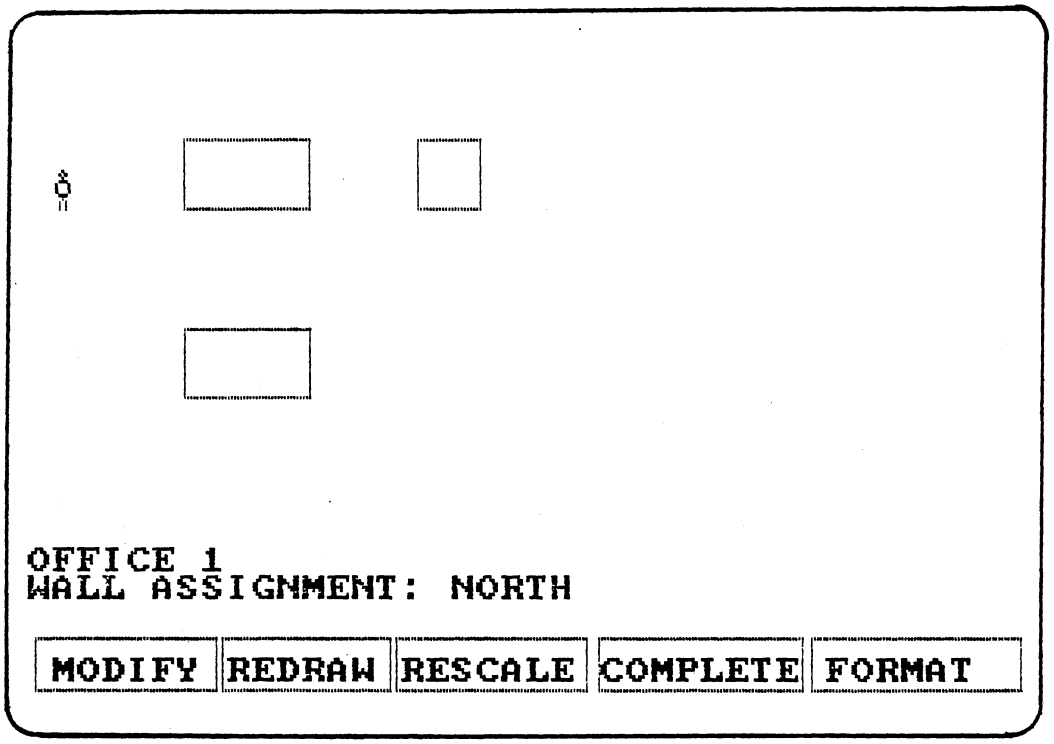
After the physical characteristic data is entered, CREATE draws a plan view, a front elevation and a side elevation of the activity. The designer may now interact with CREATE and modify the shape or size of the displayed activity. As shown in Figure 11, the designer may modify the form of the activity by changing the shape of the plan view. Using the light pen, the designer can add walls, delete walls or just change the length of an existing wall. All changes to the plan view are automatically reflected in the elevation views. During modification, building lines are sometimes erased and the display is difficult to distinguish. When this occurs, the user may select the REDRAW option which redraws the three activity views.

The other modification options available in the SPACE routine are SCALE and FORMAT. SCALE allows tbe user to change the scale of the activity views. CREATE first displays the current scale and then prompts the designer to input the desired scale from the keyboard. FORMAT allows the user to change the physical characteristics information describing the activity. CREATE returns to the input screen and after the data is entered, new activity plan and elevation views are drawn.

When the relationships between activities have been established and the physical shape of each activity has been determined, the designer is ready to order the activities. The ordering scheme revolves around four selected circulation patterns. CREATE next calls for the CIRCULATION routine.

CIRCULATION Routine

The CIRCULATION routine contains the circulation pattern, or ordering routine menu board. The major circulation development routines are



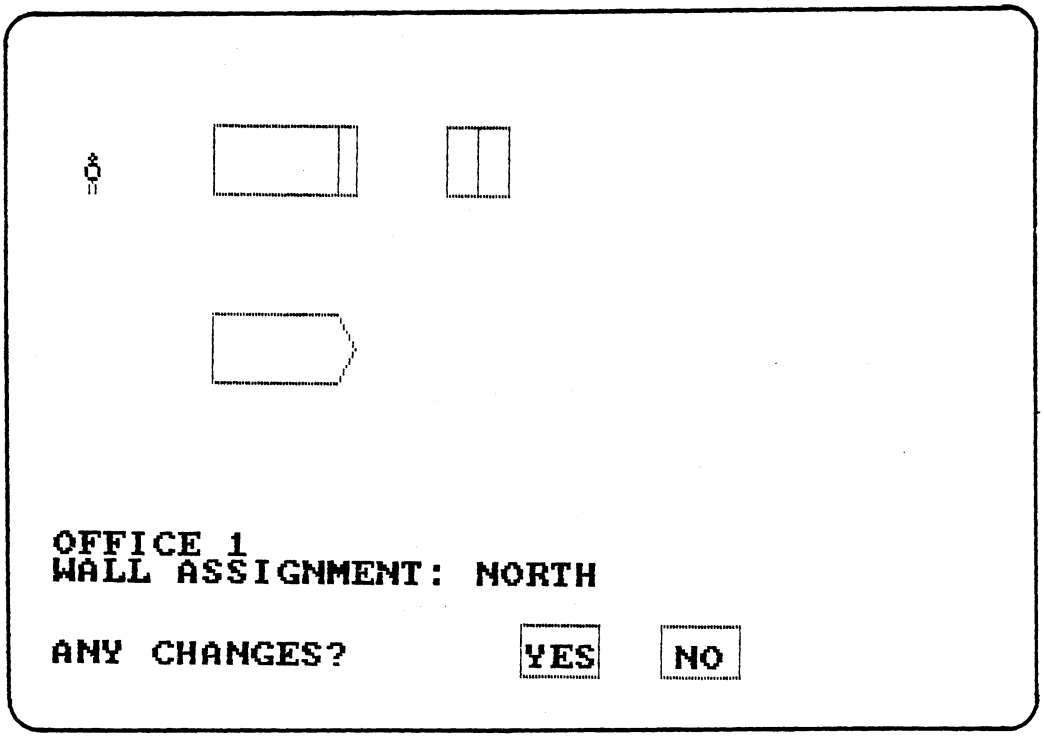


Figure 11. SPACE Routine Activity Display Screens

SPINE, COURTYARD, RADIAL, and CLUSTER. As shown in Figure 12, the user indicates his selection with the light pen. CREATE then calls the appropriate routine into memory.

SPINE uses a line type circulation pattern. Activities are placed on either side of the spine. The spine's length and width are user defined variables.

COURTYARD uses a point type circulation ordering scheme. The user specifies the courtyard dimension and activities are placed adjacent to the outside of the courtyard perimeter.

RADIAL is a variation of the point type ordering pattern. The designer inputs the courtyard dimension, the hallway width and the hallway length. Activities are then placed both inside and outside the courtyard perimeter along the designated hallways.

CLUSTER allows the designer to create his own ordering scheme. He first specifies a grid pattern size, which is used as a background guide and then, using the light pen, indicates where each activity should be placed.

Each of the circulation routines has specific placement guidelines. The next four sections discuss the information and logic flow of each technique.

SPINE Routine

The SPINE routine places activities along the north and south perimeter of a linear circulation pattern. Figure 13 illustrates the logic flow through the SPINE routine. Currently, the SPINE routine allows only one circulation pattern to be used, however, future plans will include alternative patterns, such as an offset spine, an angled spine and a curved spine.

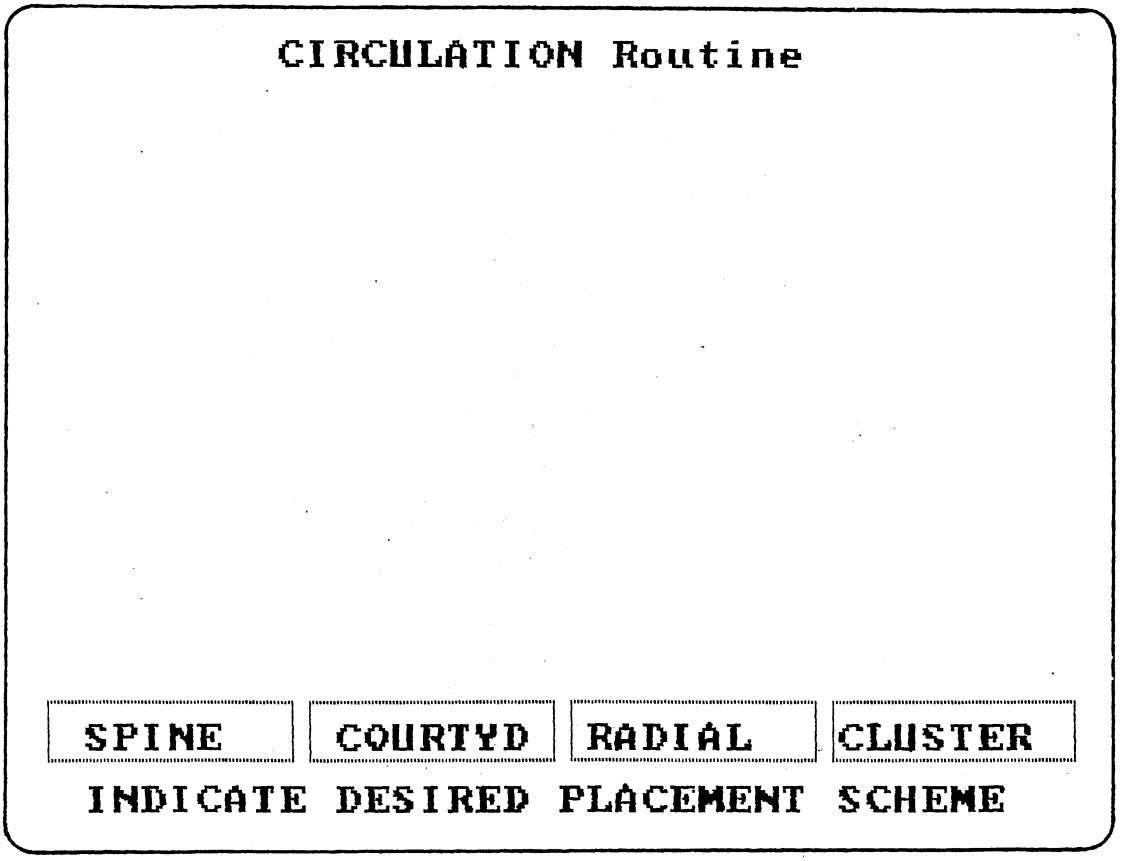


Figure 12. CIRCULATION Routine Menu Board

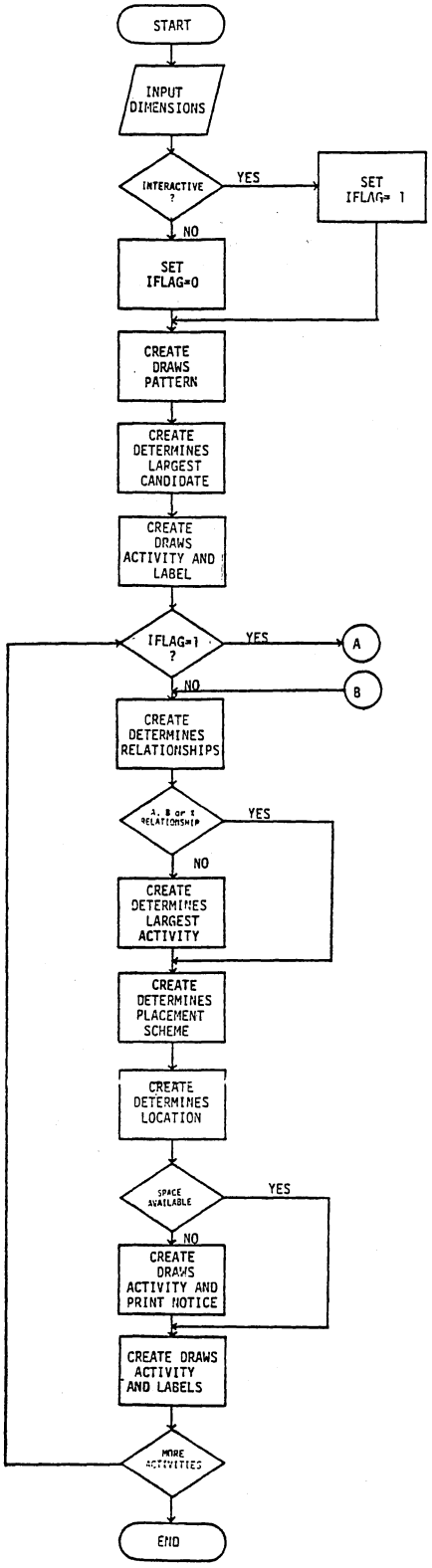


Figure 13. Placement Routines Flow Diagram

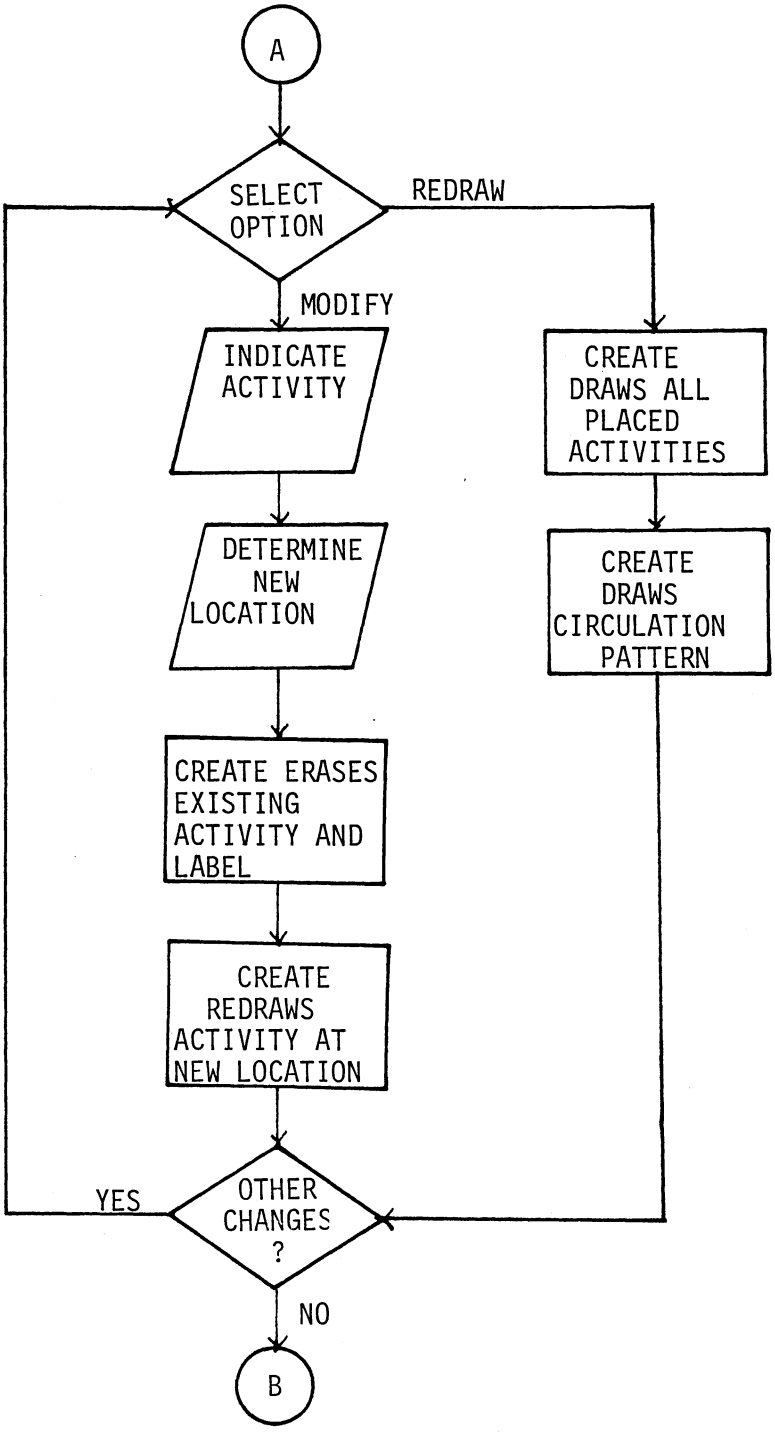


Figure 13.(Continued) Placement Routines Flow Diagram

Initially, CREATE prompts the user for three data entries, the spine width, the spine length and whether or not the user desires the interactive mode. In the interactive mode, the user must approve each activity placement and has the ability to modify the placement location of any placed activity. To modify an activity placement, the designer, using the light pen, selects the MODIFY option and indicates the activity to be moved. CREATE then determines which activity is being specified. The user is then prompted to indicate the new placement location, again using the light pen. Once indicated, CREATE determines on which side of the spine the activity will be located and translates the selected activity's corner points to the new spine location. The current activity display and label are erased and the new activity plan view drawn.

Once the designer has input the initial data, CREATE draws the spine on the screen. Next, CREATE determines which activity has the largest square footage requirement. Square footage is used as a design decision criteria to decrease the chances of an irregularly shaped building, as it is easier to arrange small spaces around a large space than vice versa.

After selecting the largest activity, CREATE next determines the wall assignment of this initial activity. In the SPINE routine activities with a north or east wall assignment are initially placed on the north perimeter, with the lower right hand corner of the activity located at the spine midpoint. South and west activities have an initial position on the south perimeter, with the upper right activity corner located at the spine midpoint. If no wall assignment is indicated, CREATE initially defaults to the north perimeter.

CREATE places the initial activity at the proper position by drawing lines between the activity's corner points. As previously

described, if the interactive mode is in effect, the user must approve the placement of the activity. The corner points of each activity are stored in a three dimensional linked list data structure. In addition, CREATE also remembers which corner points are adjacent to the spine perimeter. The structure of both arrays are shown in Table V.

Referring to Figure 13, CREATE next scans the relationships between placed activities and all unplaced or candidate activities. CREATE first determines if any "X" relationships exist. If so, that candidate activity is identified and the wall assignment determined. CREATE reserves the spine's corner positions for the placement of candidate activities having "X" relationships with placed activities. CREATE attempts to satisfy the candidate activity's wall assignment requirement, although separation is given higher priority. For example, suppose a candidate activity has a north wall assignment and an "X" relationship with a placed activity. If the placed activity is located on the north perimeter, CREATE will override the candidate activity's wall assignment and place it on the south perimeter to insure separation. The user may then modify if desired.

If no "X" relationships with placed activities are present, CREATE determines the largest numerical relationship rating between placed and candidate activities. If this score is greater than or equal to a "B" relationship, that activity is selected to be placed next on the spine. If the numerical score is less than a "B" relationship CREATE then determines which of the candidate activities has the largest square footage requirement and that activity will be placed next on the spine.

TABLE V

DATA STRUCTURE SUMMARY

Array Elements	Description
PLANPTS (A,B,C)	Activity corner point information where A - activity identification B - corner point identification C - flag value where if C=1 - header value C=2 - X-coordinate C=3 - Y-coordinate C=4 - pointer value
TRACK (A,B)	Spine adjacency information where A - selection and placement sequence B - flag value where if B=1 - activity identification B=2,4,6,8 - X-coordinates of corner points adjacent to spine Ø otherwise B=3,5,7,9 - Y-coordinates of corner points

Placement on the spine is based on a number of guidelines and rules. The first step is to determine which of the sixteen placement schemes is required. Choice of placement scheme is dependent upon the wall assignments of the placed and candidate activities. For example, if the placed activity is located on the north perimeter of the spine and the candidate activity has an east wall assignment, CREATE uses rule set 10 to determine the proper placement of the incoming activity. Each of the sixteen schemes has a set of rules used to guide placement.

All rule sets follow a similar pattern. Using the example above CREATE first determines whether or not the candidate activity may be

placed to the east of the placed activity without extending past the spine end point (Corner points are reserved for X-relationships.) If the candidate activity can be placed without overlapping into the reserved area, CREATE determines if there is sufficient room to place the candidate without overlapping a previously placed activity. If both guidelines are satisfied, CREATE then places the activity on the spine. If either guidelines cannot be satisfied, CREATE then conducts another series of tests.

If a candidate activity has an east wall assignment, CREATE will not place it to the west of the spine midpoint. Likewise, an activity with a west wall assignment may not be placed to the east of the spine midpoint. Referring to the example above, since the candidate activity failed the first series of tests, instead of checking the west side of the placed activity, CREATE determines if the candidate activity may be placed east of the midpoint on the south perimeter of the spine. CREATE determines if there is sufficient space to locate the candidate activity without overlapping a previously placed activity or extending into the space reserved for activities having "X" relationships. If so, the activity is placed on the spine. Figure 14 illustrates the SPINE layout process. Table VI contains the data used to develop the spine layout, as well as the other three circulation pattern layout plans. As discussed previously, there are sixteen different placement schemes and each one has guidelines tailored for its own particular application. The rule set for the SPINE routine is shown in Appendix B.

TABLE VI

INPUT DATA FOR EXAMPLE LAYOUT DEVELOPMENT PROCESSES

Activity Name	Area	Proportion	Ceiling Height	Wall Assignment
OFFICE 1	250	1 to 2	10	North
OFFICE 2	250	1 to 2	10	North
OFFICE 3	250	1 to 1	10	East
RECEPTION	400	2 to 1	15	South
FILE ROOM	200	1 to 2	15	West
CONFERENCE	500	1 to 2	12	West

Activity	vs	Activity	Relationship
OFFICE 1		OFFICE 2	A
OFFICE 1		OFFICE 3	A
OFFICE 1		RECEPTION	B
OFFICE 1		FILE ROOM	C
OFFICE 1		CONFERENCE	B
OFFICE 2		OFFICE 3	A
OFFICE 2		RECEPTION	B
OFFICE 2		FILE ROOM	C
OFFICE 2		CONFERENCE	B
OFFICE 3		RECEPTION	B
OFFICE 3		FILE ROOM	C
OFFICE 3		CONFERENCE	B
RECEPTION		FILE ROOM	X
RECEPTION		CONFERENCE	B
FILE ROOM		CONFERENCE	C

If CREATE cannot satisfy the guidelines, it then draws the candidate activity's plan view in the lower right hand corner of the screen and displays the candidate's relationship with the placed activity, as shown in Figure 15. The user must then indicate with the light pen, the desired location position on the spine and allows the user to make modifications to the layout design.

To place an activity on the spine, CREATE translates the activity's corner points to specified spine locations. Using the new corner points, CREATE draws the activity on the spine. Activities are not allowed to

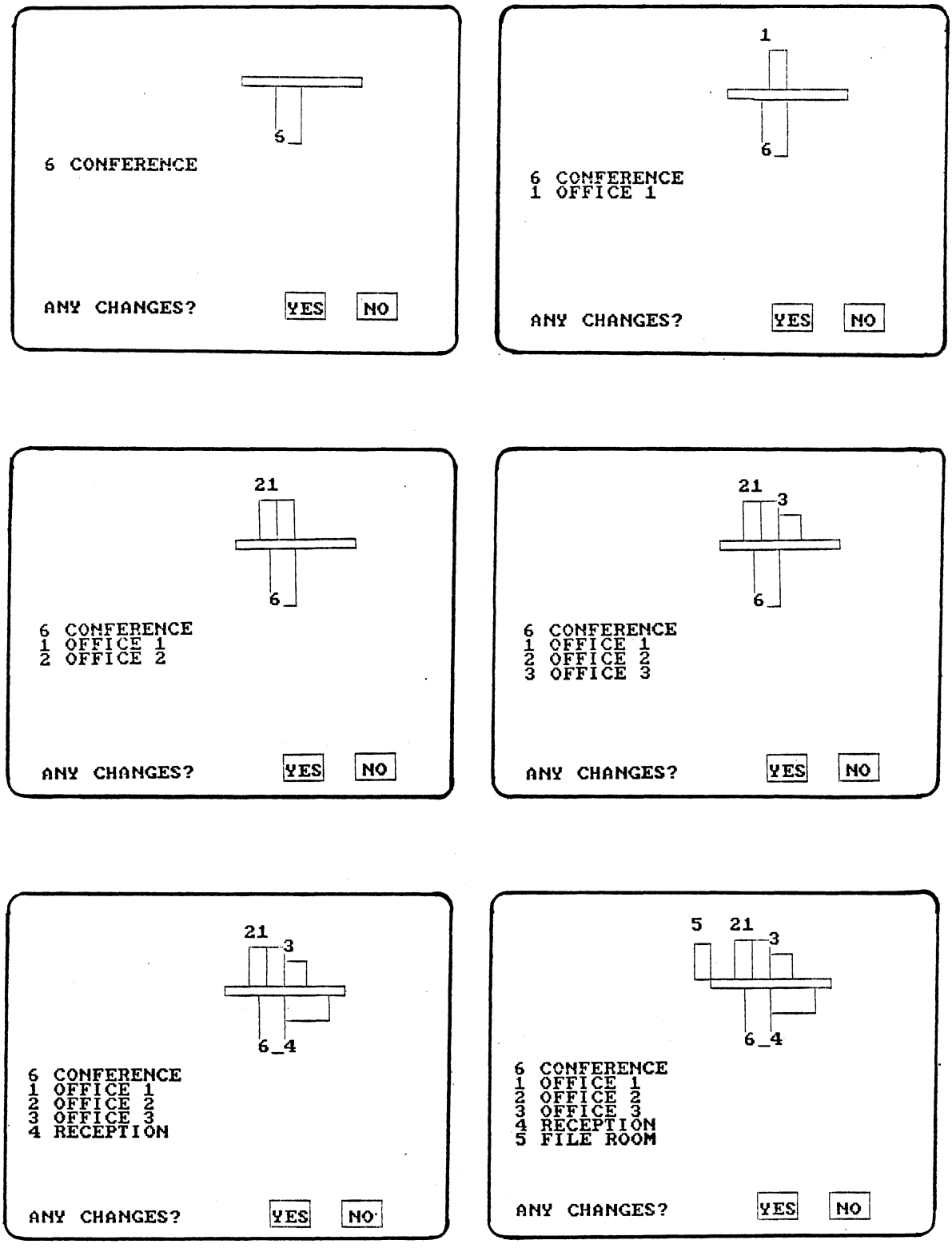


Figure 14. SPINE Routine Layout Development Process

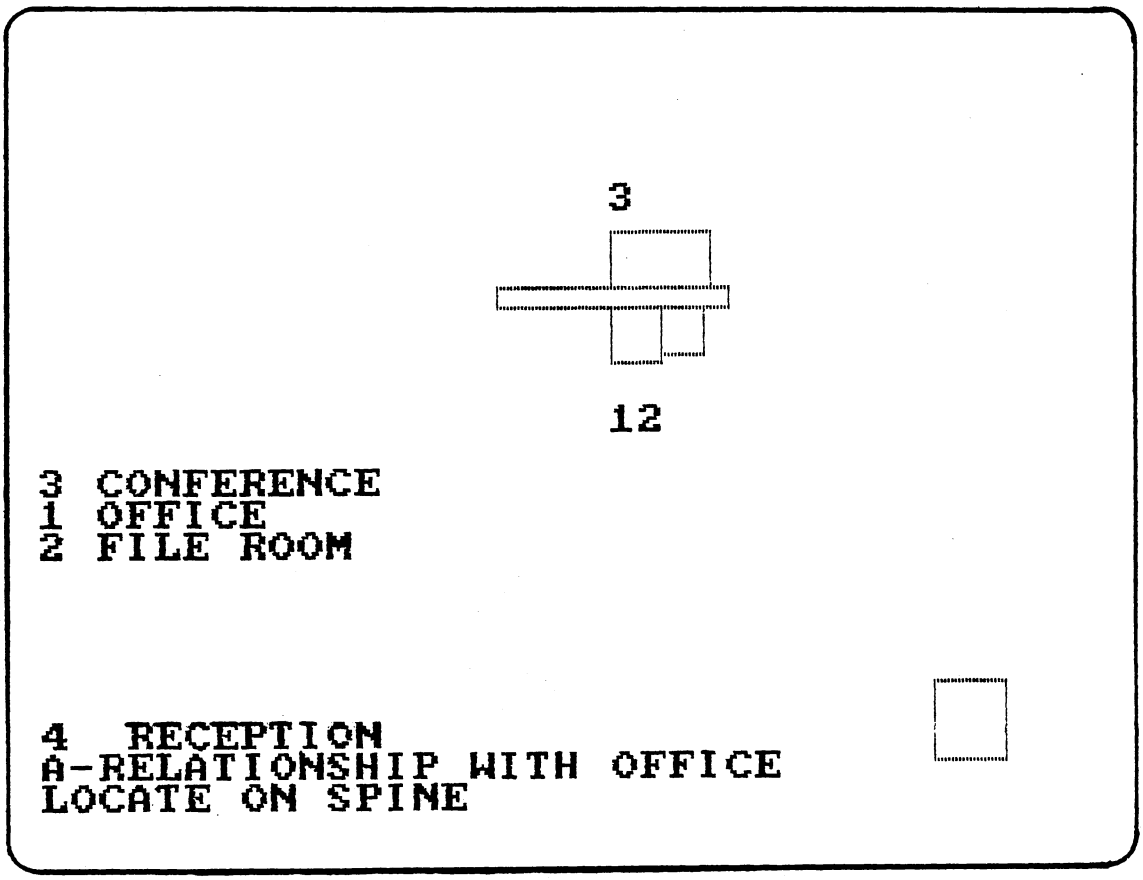


Figure 15. Layout Development Default Screen

overlap and must be adjacent to the spine perimeter. After constructing the activity, CREATE then displays an identification number and prepares a legend below the plan drawing. The legend is in order of placement.

When all activities have been placed on the spine, the user may select from several options including save the design, modify the existing design, create another design, create a site plan, or begin new problem.

COURTYARD Routine

The COURTYARD routine places activities around the perimeter of a square courtyard circulation pattern. The guidelines used to determine the design of the courtyard plan are similar to those outlined in Figure 13 and described in the SPINE routine section. As with SPINE, CREATE allows only one courtyard circulation pattern, however, additional patterns should be included in the future.

To begin the COURTYARD routine, CREATE prompts the user for the courtyard dimension, and as with the SPINE routine, the user is given the choice of interactive capability. Once these initial inputs are processed, CREATE draws a square courtyard on the screen using the inputted courtyard dimension.

CREATE next selects the activity with the largest square footage requirement, checks that activity's wall assignment and locates the activity on the courtyard. All activities are placed on the outside of the courtyard perimeter and the courtyard area is designated as the circulation space. The placement process of the COURTYARD routine differs from the SPINE routine. In the SPINE routine, CREATE was programmed to place activities on either the north or south perimeters of the spine circulation pattern. East and west activities were

located east or west of the spine midpoint, but only two perimeter edges were used for placement. In the COURTYARD routine, activities may be placed on all four sides of the circulation pattern. The design is developed in a clockwise fashion. For example, if the initial activity has a north wall assignment, CREATE will place it at the left corner of the north perimeter. Similarly, if the activity had an east wall assignment, it would have been placed on the east perimeter at the upper right hand corner of the courtyard.

CREATE then determines if any of the candidate activities have an "X" relationship with a placed activity. As with the SPINE routine, the corner locations of the courtyard are reserved for candidate activities that require separation from placed activities. If no "X" relationships occur, CREATE determines which candidate activity has the largest numerical relationship with the placed activities. If at least a "B" relationship exists, CREATE then determines the placement location of the candidate activity. As with the SPINE routine, if no "A" or "B" relationship exists, CREATE determines which of the candidate activities has the largest area and selects it for placement on the courtyard.

As stated previously, placement is clockwise. As with the SPINE routine, CREATE first determines which of the sixteen placement schemes is required. Because placement is clockwise, CREATE initially places activities adjacent to the corner positions and checks for available space on only one side of the placed activity. For example, if the placed activity is on the north perimeter and the candidate activity has a north wall assignment, CREATE will determine if space is available to the east of the placed activity. Similarly, if the placed and candidate activities both have east wall assignment, CREATE will check

for available space south of the placed activity. If space is available, the candidate activity is placed in the courtyard. Figure 16 illustrates the placement scheme for the COURTYARD routine. Appendix C contains the placement guidelines.

Construction of the line drawings for the COURTYARD routine are the same as the SPINE routine. The corner points of the activity are translated to the courtyard perimeter and lines drawn between corner points. After completing the drawing the user has the same options available at the end of the SPINE routine, one being to select another circulation pattern and create another design.

RADIAL Routine

The radial circulation pattern is the most complex pattern offerred in CREATE. The currently programmed radial ordering system is the pinwheel pattern. Figure 17 illustrates the pinwheel pattern and describes the wall assignment notation. To begin the RADIAL routine, the user must enter the inner courtyard dimension, the hall width and the hall length. As with all of CREATE's circulation pattern alternatives, the user is also prompted for use of the interactive mode.

The placement guidelines used in RADIAL are similar to those programmed in both SPINE and COURTYARD. The major differences occur when CREATE determines the actual placement location of a candidate activity. As shown in Figure 17, activities may be placed on the inside perimeter of the courtyard, along the outside perimeter of the courtyard hall and adjacent to the wing halls. In addition, as with both of the previous ordering patterns, activities with "X" relationships are reserved for the outer corner locations.

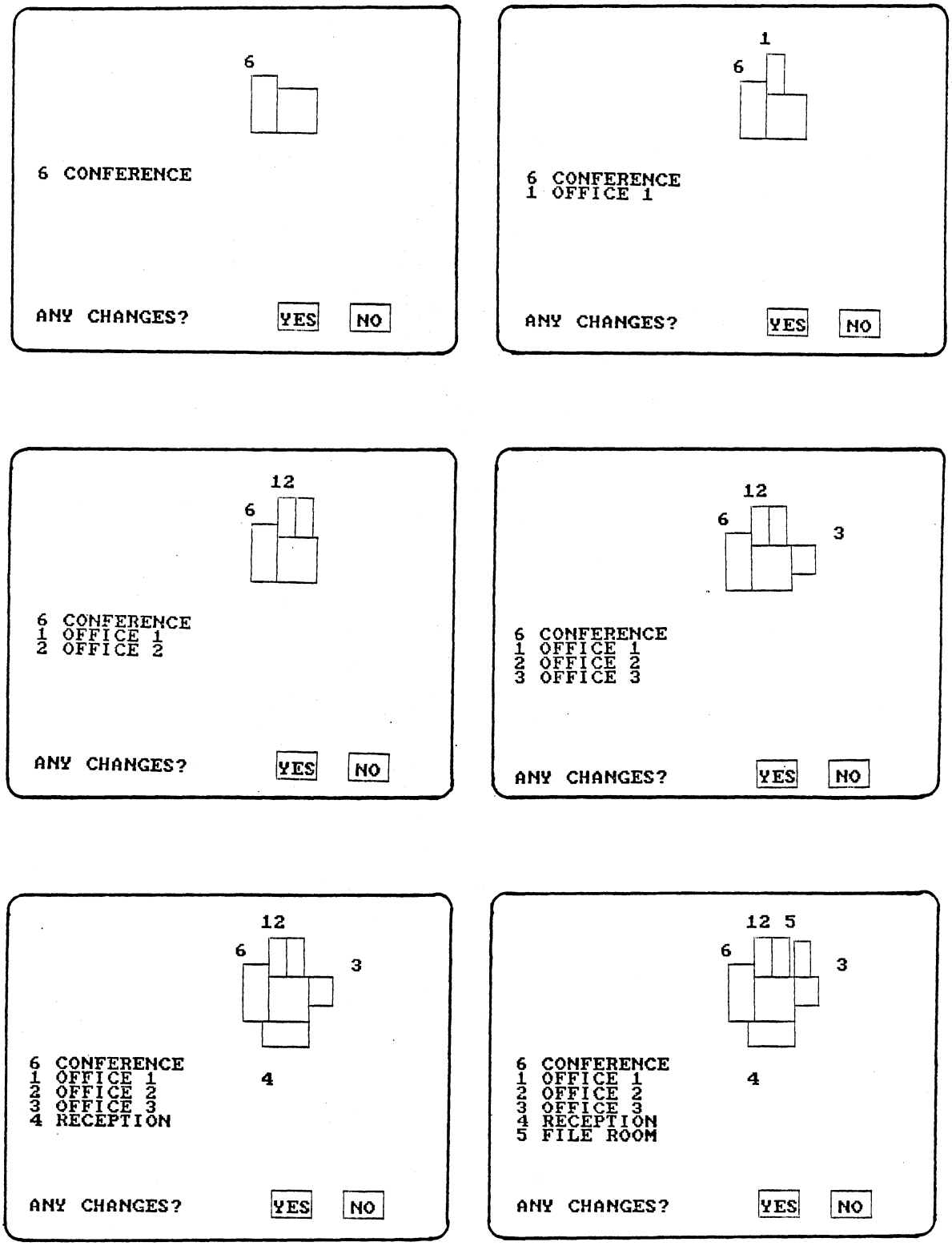


Figure 16. COURTYARD Routine Layout Development Process

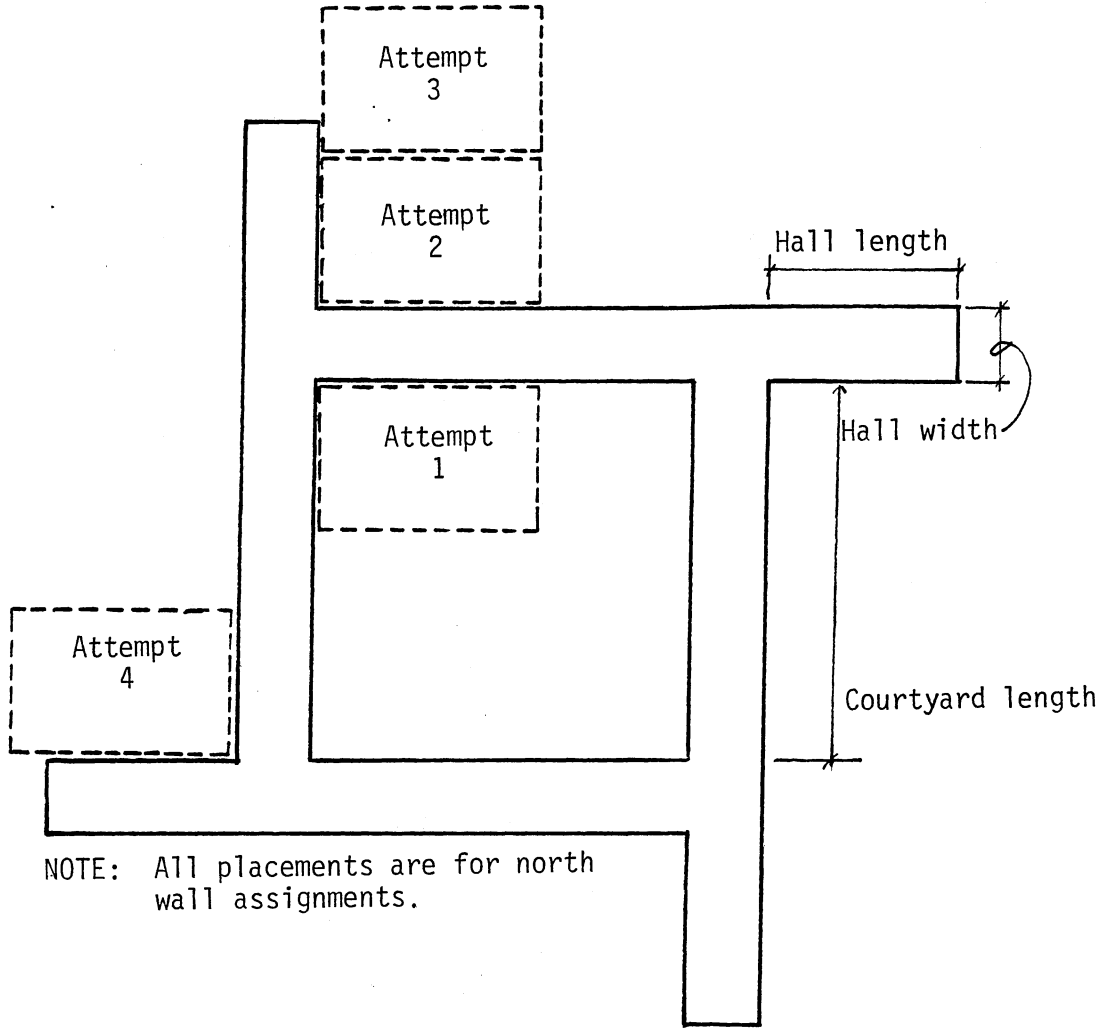


Figure 17. RADIAL Circulation Pattern

The RADIAL placement process is a combination of the guidelines used in the SPINE and COURTYARD routines. After selecting the activity with the largest area, and checking that activity's wall assignment, CREATE determines if the activity can be positioned within the inner courtyard space without overlapping other activity boundaries or the circulation pattern hallways. If so, CREATE places the activity in the corner corresponding to the activity's wall assignment. For example, if the initial activity has a south wall assignment, CREATE will place the activity adjacent to and within the inner courtyard perimeter in the lower right hand corner of the courtyard. CREATE fills the inner courtyard in a clockwise fashion, similar to the COURTYARD routine. If the activity is too large for the inner courtyard, CREATE will place the activity directly across the courtyard hallway. Referring to the previous example, if the initial activity is too large for the inner courtyard area, CREATE will place the activity directly south of the lower right hand inner courtyard corner, adjacent to the courtyard hallway and the wing hallway.

Candidate activities are selected based on their relationships with placed activities, as described in both SPINE and COURTYARD. "X" relationships are checked first, followed by "A" or "B" relationships. In addition, the activity square footages are used as default placement criteria. The guidelines within each of the sixteen placement schemes are more complicated than either SPINE or COURTYARD because of the increased number of possible locations in the pinwheel circulation pattern. As illustrated in Figure 17, in contrast to the SPINE and COURTYARD routines, an activity with a north wall assignment can be placed on four pinwheel edges and still satisfy the wall assign-

ment requirement. The same is true for the other three wall assignment alternatives. Appendix D contains RADIAL placement guidelines.

CREATE attempts to place a candidate activity, with the same wall assignment as the placed activity, as close as possible to that placed activity. CREATE first attempts to place the candidate activity in the inner courtyard. If the space is already occupied, CREATE then tries to position the candidate activity along the outer courtyard hallway, corresponding to the activity's wall assignment. If there is not enough space along this edge, CREATE next checks the adjacent wing hall and, if space is available, places the activity. If none of these three areas have sufficient space for placement of the candidate activity, CREATE checks the opposite wing hallway, as demonstrated in Figure 17. Finally, if no placement position can be determined, CREATE defaults as described in SPINE and COURTYARD, and prompts the user to place the candidate activity on the pinwheel.

If the candidate activity does not have the same wall assignment as the placed activity, CREATE follows a placement process similar to that discussed previously by determining which of the four perimeter edges, corresponding to the candidate activity's wall assignment, have space available for placement of the candidate activity. Figure 18 illustrates the pinwheel activity placement process.

Candidate activity plan views are drawn using a process similar to that described for both the SPINE and COURTYARD routines. Activity boundaries are not allowed to overlap nor are they allowed to extend into the hallway spaces. Once complete, the designer may store the design, create another design, or create a building site and orient a building design on the site.

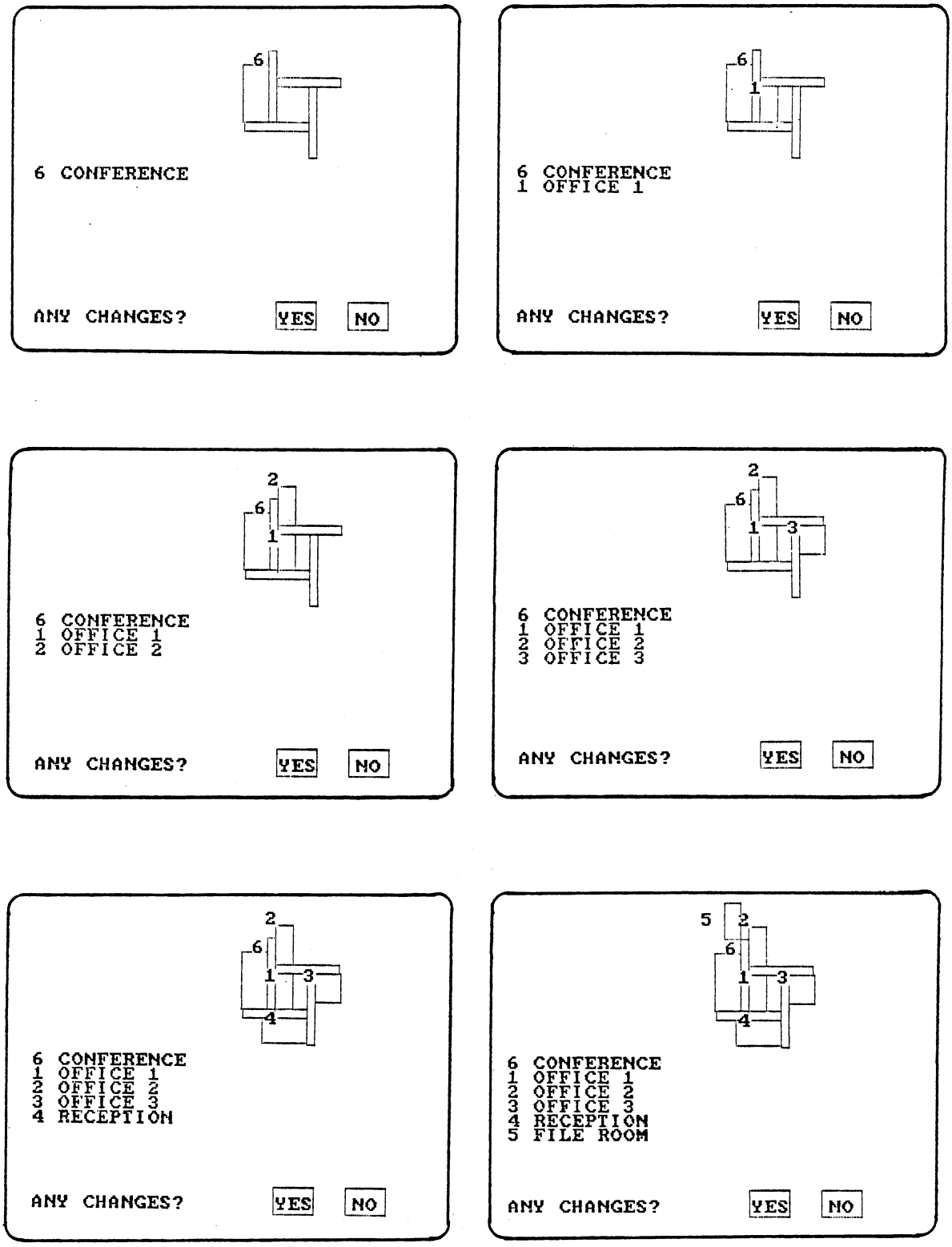


Figure 18. RADIAL Routine Layout Development Process

CLUSTER Routine

CLUSTER is the simplest design routine, but allows the greatest amount of design development freedom. The CLUSTER routine allows the designer to develop his own circulation pattern. CREATE selects the placement order of the activities but the user must determine where each will be placed in relation to one another. The CLUSTER routine begins by asking the user to input a grid size. After the information is read, CREATE draws a grid pattern using the inputted dimension.

The activity selection process is similar to the previously described circulation patterns. CREATE first determines which activity has the largest square footage requirement, draws the floor plan in the lower right hand corner of the screen and displays the wall assignment. The designer is then prompted to indicate, with the light pen, a placement position. CREATE redraws the activity at the desired gird coordinate and waits for the designer to indicate approval. If the placement does not satisfy the designer, he may move the activity to a different location.

As with the other ordering routines, CREATE next determines if there are any "X" relationships between the candidate activities and the placed activities. If so, the activity floor plan, relationship and wall assignment are displayed and the designer once again prompted to indicate a placement position. If no "X" relationships exists, CREATE displays the placement information and waits for the user to position the activity into the grid pattern.

If no "A" or "B" relationships exists, CREATE determines which of the candidate activities has the largest square footage requirement.

This activity is displayed and the user prompted for placement. This placement routine continues until all activities have been positioned. The designer may continue to modify the layout until a satisfactory concept is developed. Figure 19 illustrates the cluster design development process.

As with each of the other three circulation patterns, when complete, the CLUSTER routine allows the user to save the design, develop another design or place the design on a site plan.

SITE Routine

Often building layouts must be modified to meet site restrictions. Trees may have to be saved, zoning setbacks may constrain building orientation or grade changes may require major modifications to the building design. To the author's knowledge, no currently available computer aided technique allows the designer this important step in the design development process.

CREATE allows the user to construct a building site according to inputted specifications. Currently, the site restrictions include trees and zoning or utility setbacks, as shown in Figure 20. Future plans include grade constraints and other factors restricting building placement. Figure 21 contains a flow diagram describing the SITE routine.

To begin the SITE routine. CREATE allows the user a choice of developing a new site plan or using one developed earlier and stored on disk. To enter a stored site plan the user simply indicates using the light pen, his desire to do so and enters the file name. CREATE reads the file and draws the stored site plan on the screen.

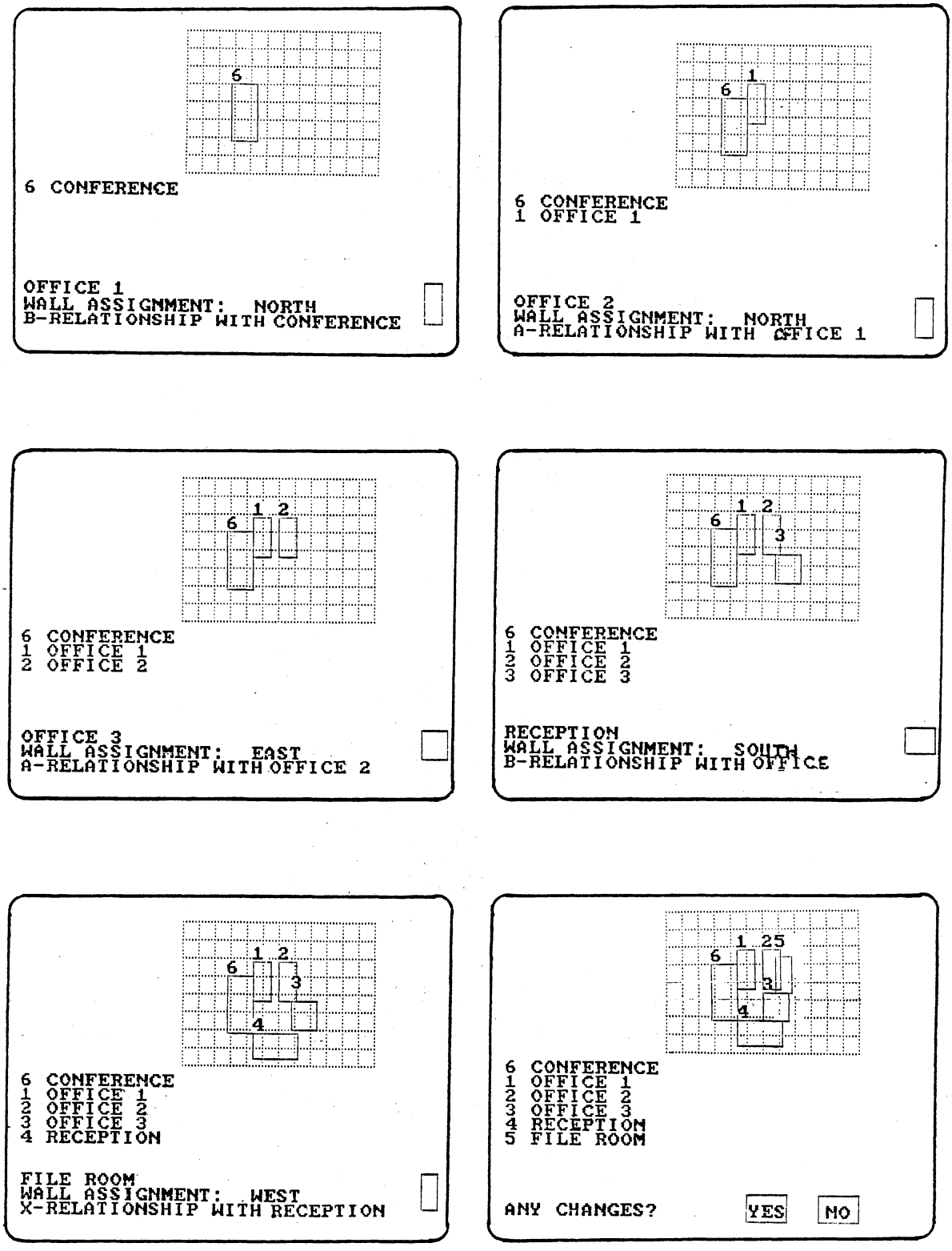


Figure 19. CLUSTER Routine Layout Development Process

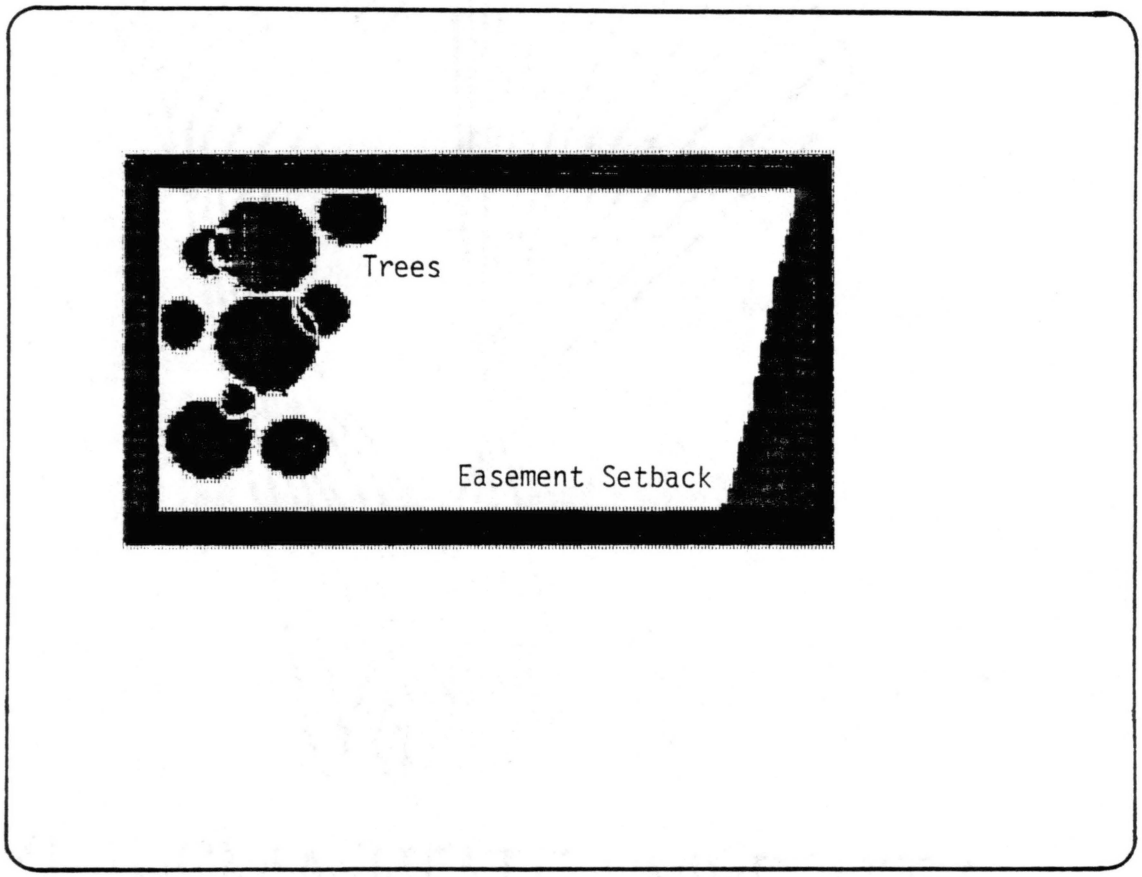


Figure 20. Typical SITE Routine Layout

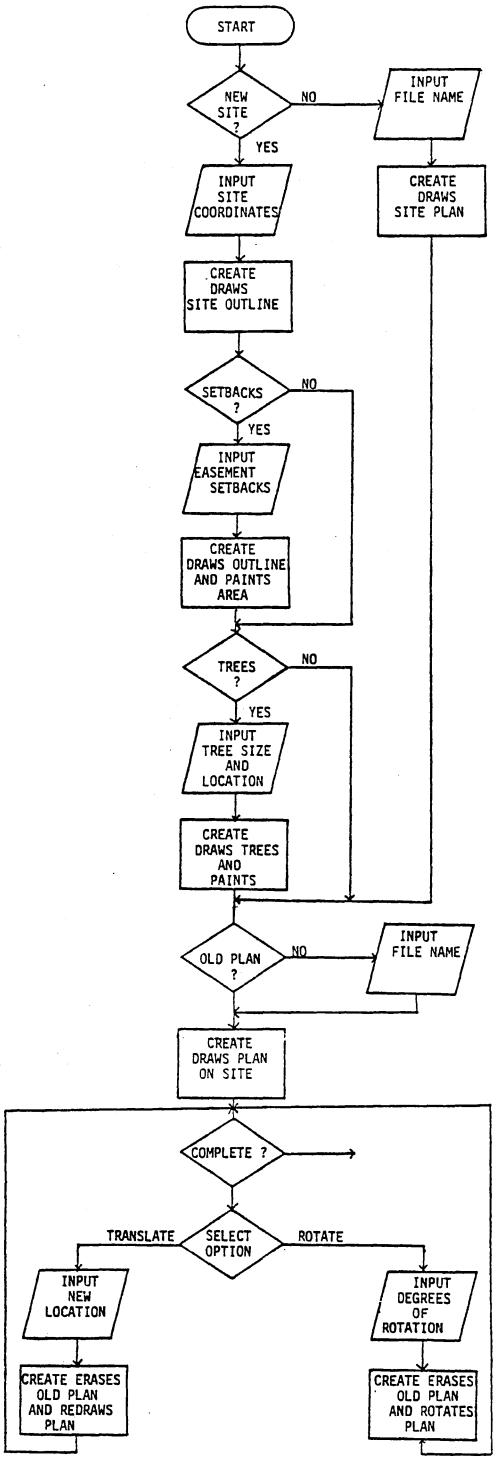


Figure 21. SITE Routine Flow Diagram

Development of a new site plan begins with CREATE prompting the designer for the building site corner points. Once all points are entered, CREATE draws the site boundary and asks if easement setbacks exist. If so, the user is prompted for the easement setback corner points. CREATE next draws the easement boundary and "paints" white the area between the site boundary and the easement boundary. This white area represents the setback space and the designer may not extend the building plan into the area. Next, or if no easement setback was necessary, CREATE allows trees to be placed on the site. If trees exist, the user, with the light pen, indicates the tree location. CREATE then prompts the designer for the tree size, a circle is drawn representing the tree and painted white indicating an area which must be avoided when positioning the building plan on the site.

After developing the site plan, the designer is ready to place a building plan onto the site. CREATE allows the user to place a previously stored plan or to place the building plan currently being developed. Once selected, CREATE draws the building plan on the screen and displays a menu board as shown in Figure 22.

The options available to the designer are rotate and translate. For example, using the light pen, the designer selects the rotate option. After prompting the user for the number of degrees rotation desired, CREATE rotates the entire building plan. The existing layout is first erased and the modified plan redrawn. Circulation patterns and activity plan views are rotated using sine and cosine functions.

Likewise, to translate the entire building form from one location to another, the designer first selects the translate option and CREATE

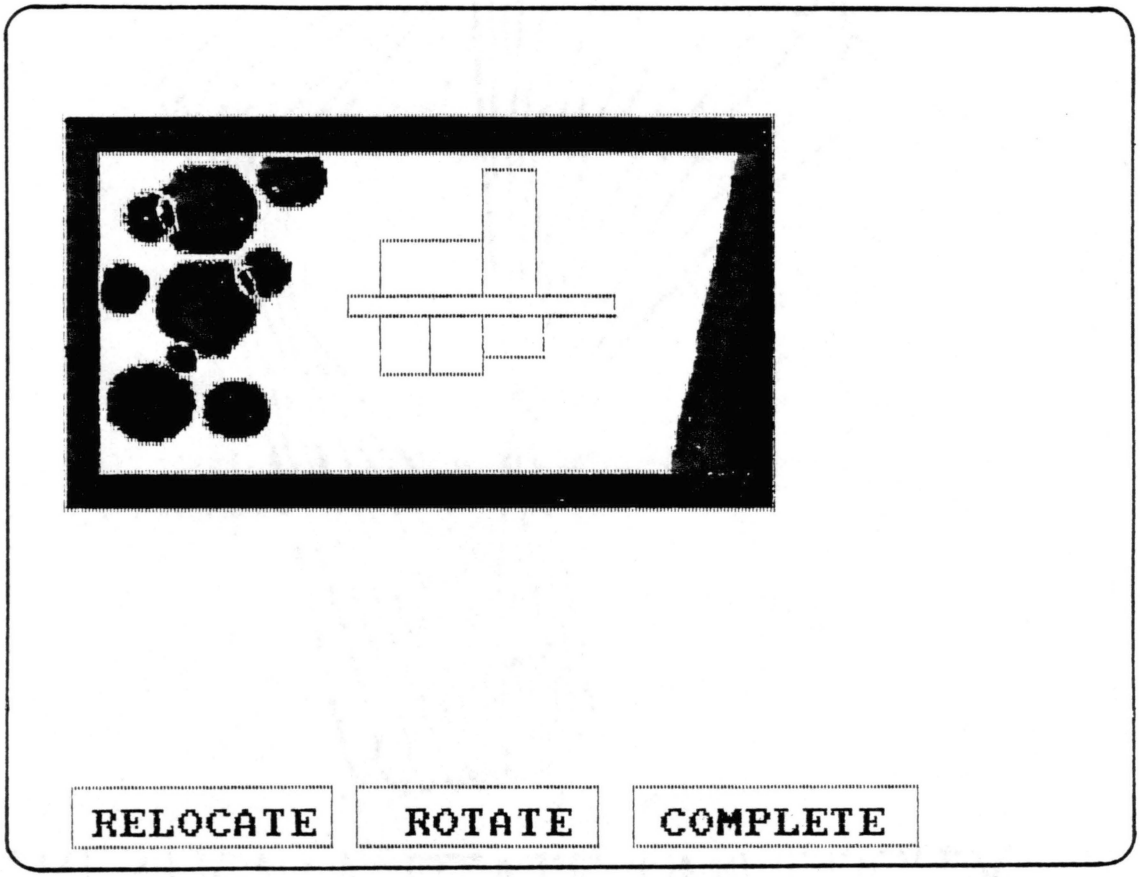


Figure 22. SITE Routine Menu Board

prompts the user for a new building location. Using the light pen, the user touches the screen to indicate the new location position. CREATE then erases the existing boundaries, translates all circulation pattern and activity corner points to their new locations and redraws the building layout.

If the building plan must be modified because of a site restriction, the designer may call a routine labelled PICTURE and modify the building shape to accommodate the site constraints. Discussed in a later section, once the building plan has been modified, the designer may once again place it on the site and repeat the development process.

Finally, if the designer is interested in viewing the building elevations, he may call the ELEVATION routine. ELEVATION, like PICTURE, allows the designer to modify the building shape and then reposition the building layout on the building site.

ELEVATION Routine

Many times designers need to see how the building plan appears in elevation. He may rearrange activities based on form, overriding the initial positioning based entirely on function. A flow diagram illustrating the ELEVATION routine is shown in Figure 23. To begin the ELEVATION routine, the user enters the desired elevation view. Then, with the light pen, the designer indicates on the building plan, which activities will be included in the elevation view. CREATE then clears the screen and constructs the selected elevation view, as shown in Figure 24. A scale figure is displayed to the left of the building elevation as a reference.

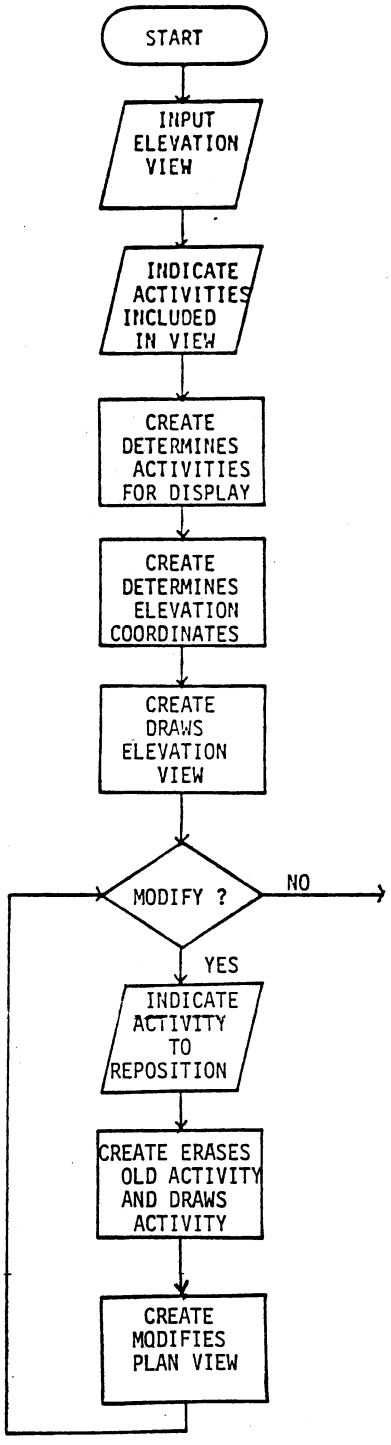


Figure 23. ELEVATION Routine Flow Diagram

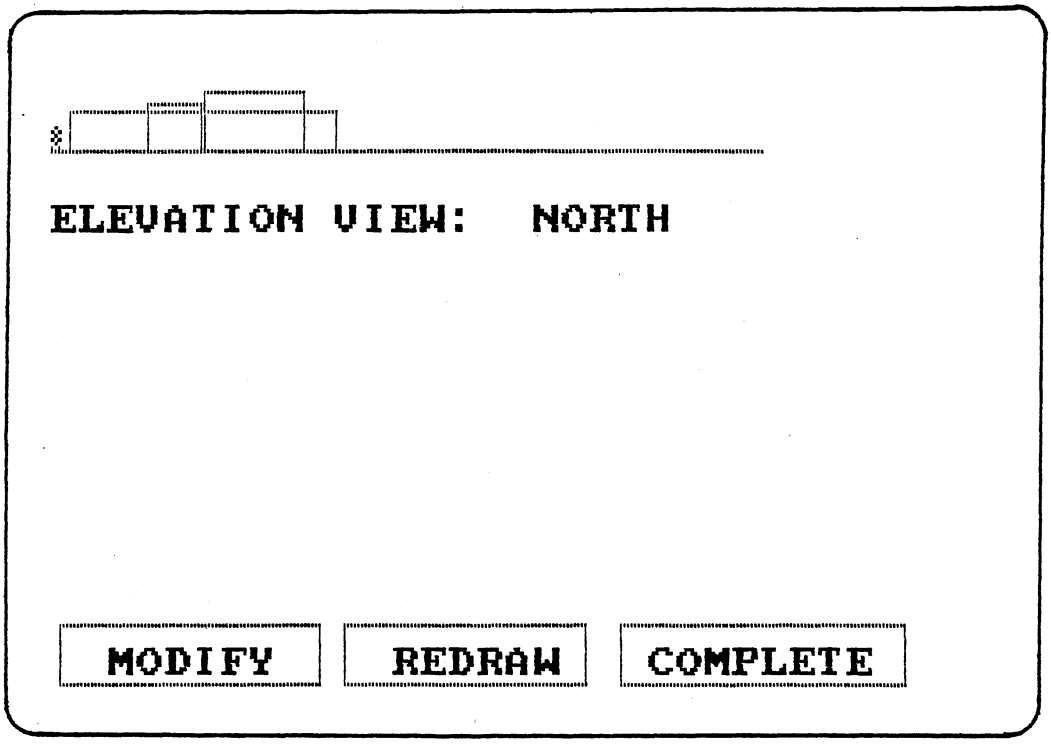
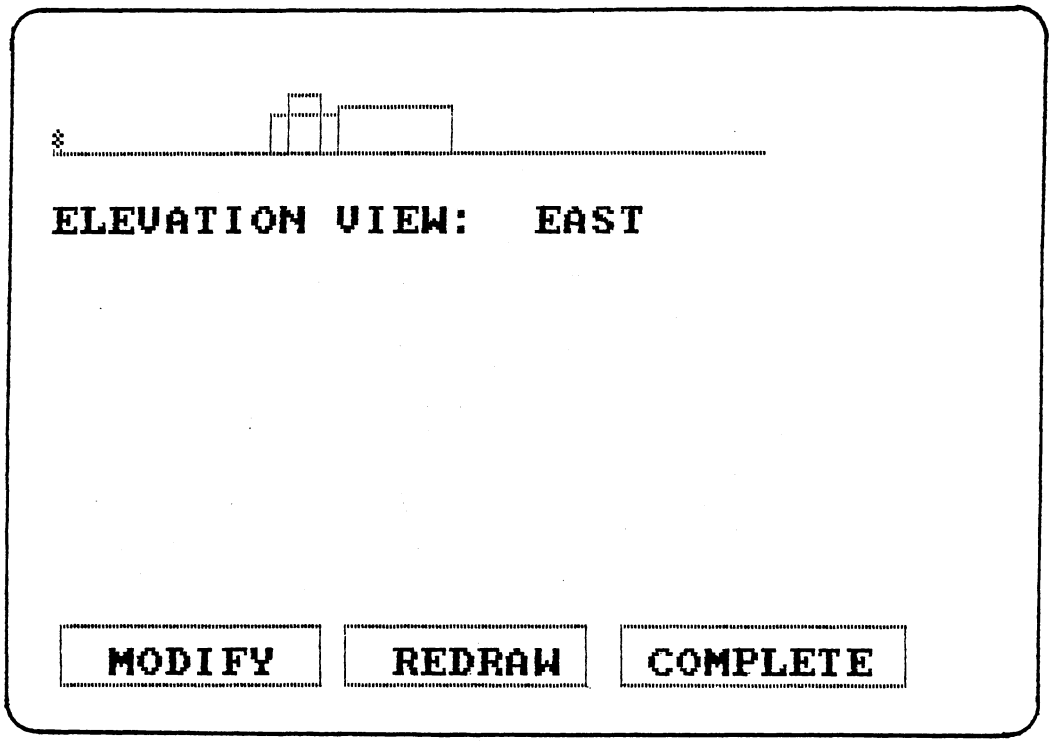


Figure 24. Elevation Views



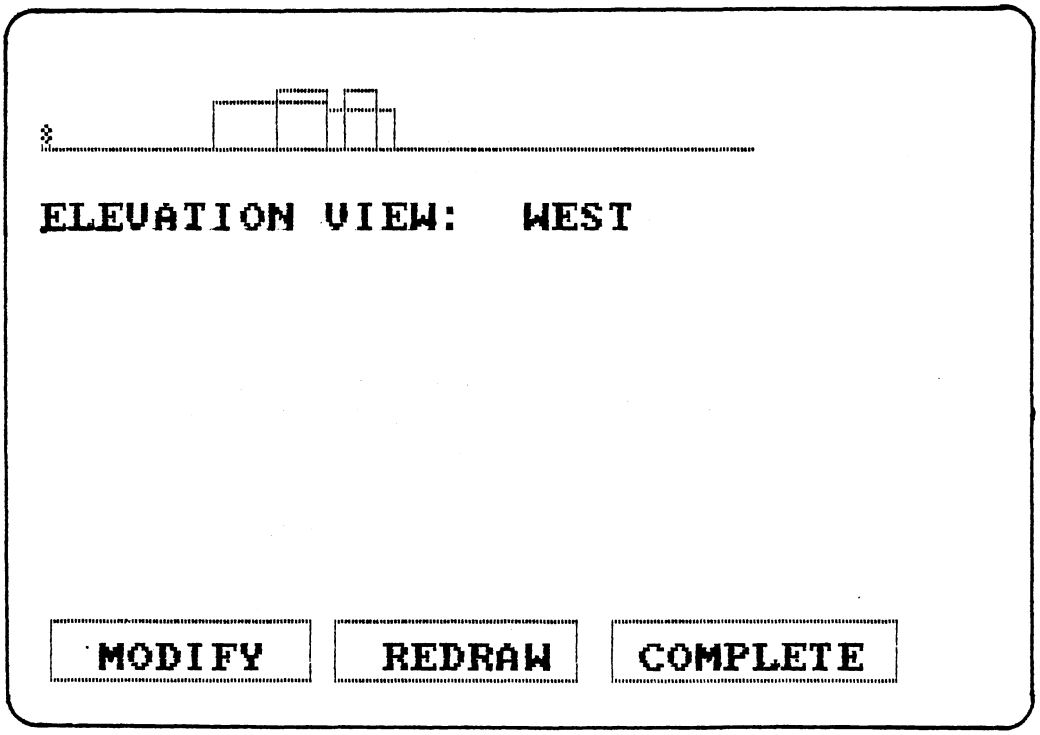


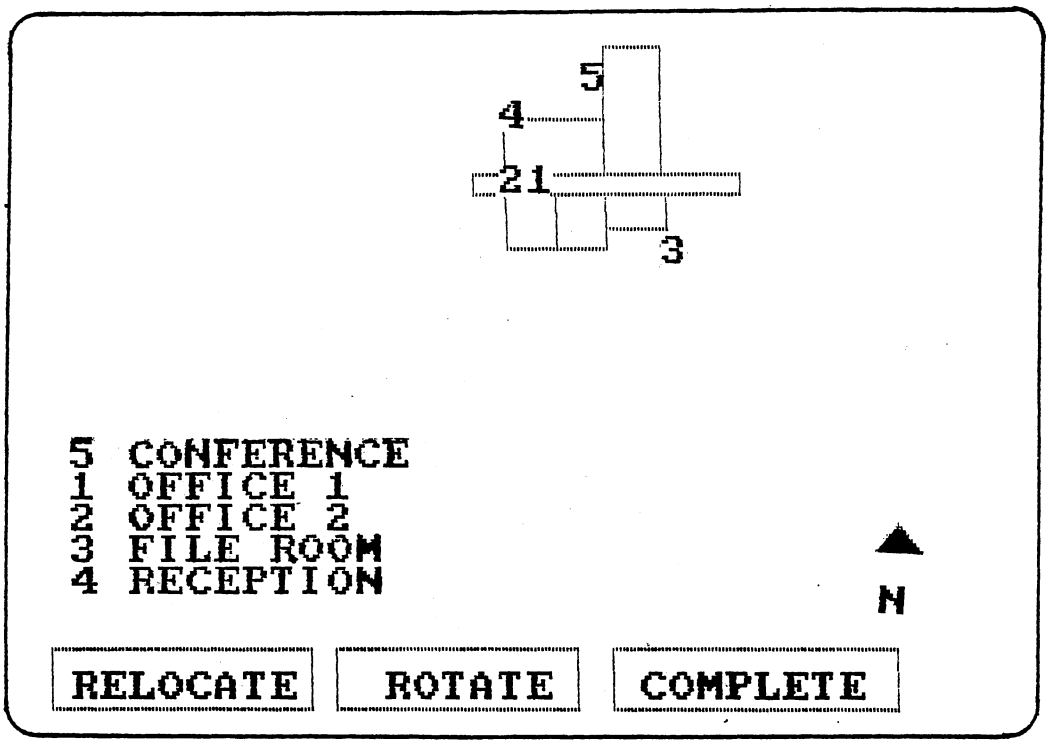
Figure 24.(Continued) Elevation Views

To construct the elevation view, CREATE first rotates, if necessary the building plan. CREATE then translates the building activities' corner points to a predetermined ground level. CREATE next, based upon the previously entered activity ceiling heights, draws the elevation view of each activity. Finally, the circulation space elevation is constructed with a preset elevation height of ten feet.

Once the elevation drawing is complete, CREATE allows the designer to modify the view based upon form preference. Using the light pen, the user indicates which activity will be translated. CREATE then erases the activity and redraws it at the new location. The building plan view is simultaneously modified to also reflect the activity location change. Figure 25 illustrates a modified elevation view and the corresponding plan view. The designer may now call the PICTURE routine to study the building and site plan and modify, if necessary, based on changes made to the elevation view.

PICTURE Routine

The PICTURE routine is used to modify both building layouts and site plans. Figure 26 outlines the flow diagram of the PICTURE routine. It can be called at the beginning of the CREATE program or after either the SITE or ELEVATION routines. As shown in Figure 27, the designer has two groups of options. The top row allows the user to translate or rotate the entire building plan. The second row options allow the user to modify the location of a specific activity or to redraw the display. The modification procedures are similar to those previously described in the ordering and site development routines.



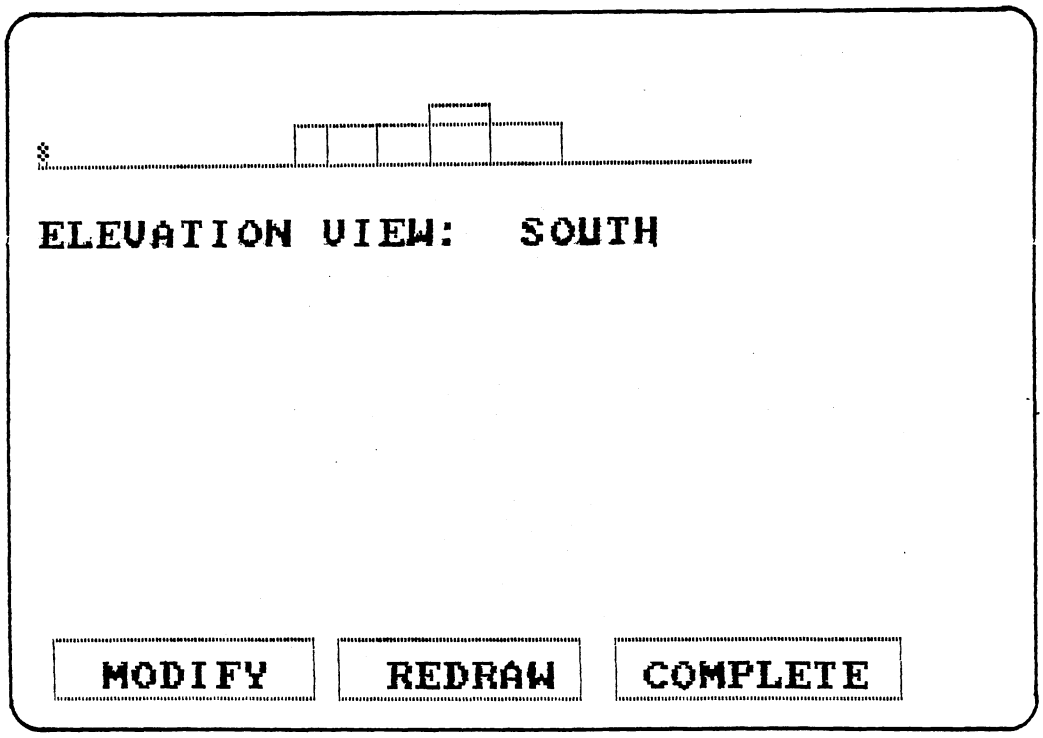
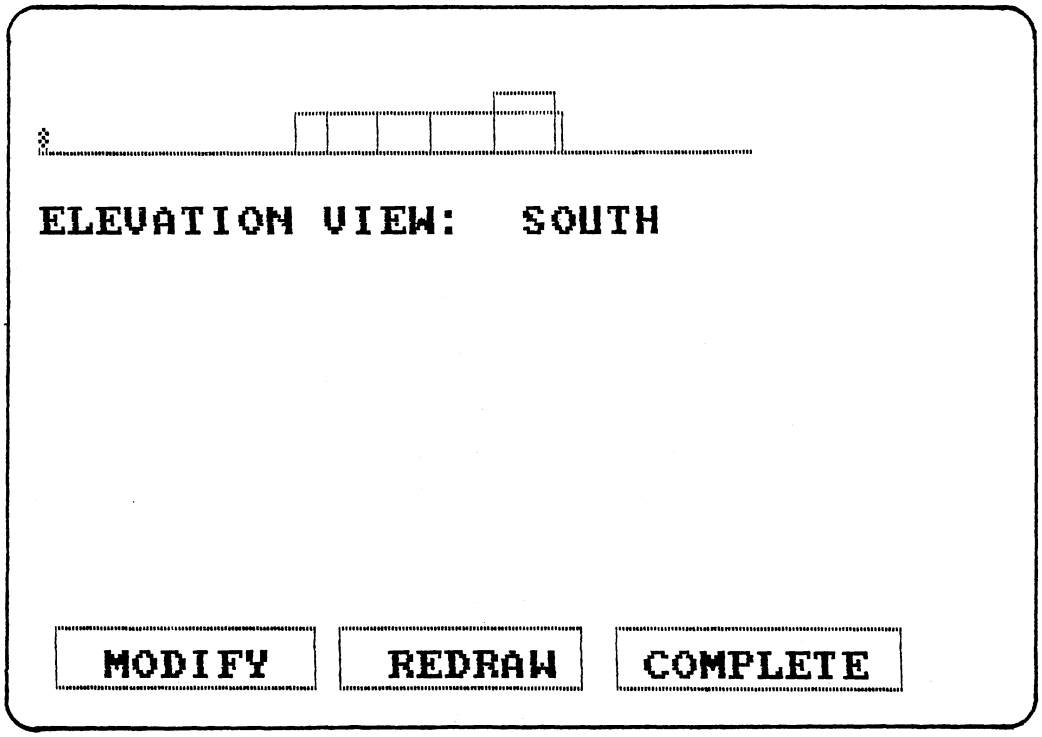


Figure 25. Modified Elevation and Plan View Process



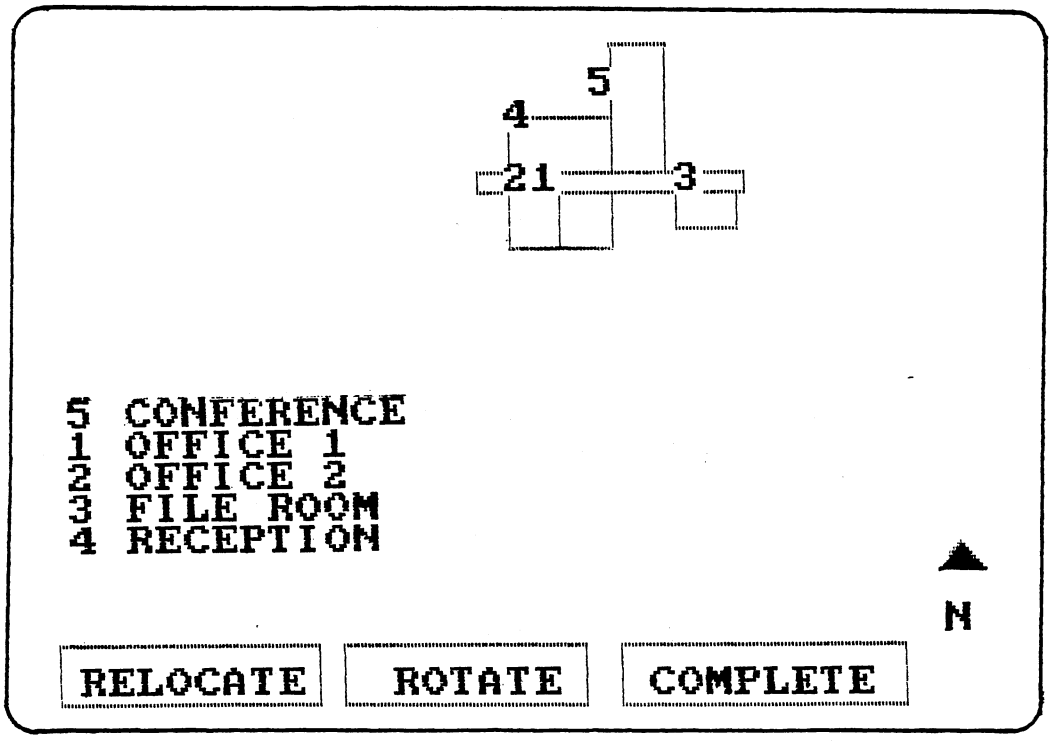


Figure 25.(Continued) Modified Elevation and Plan View Process

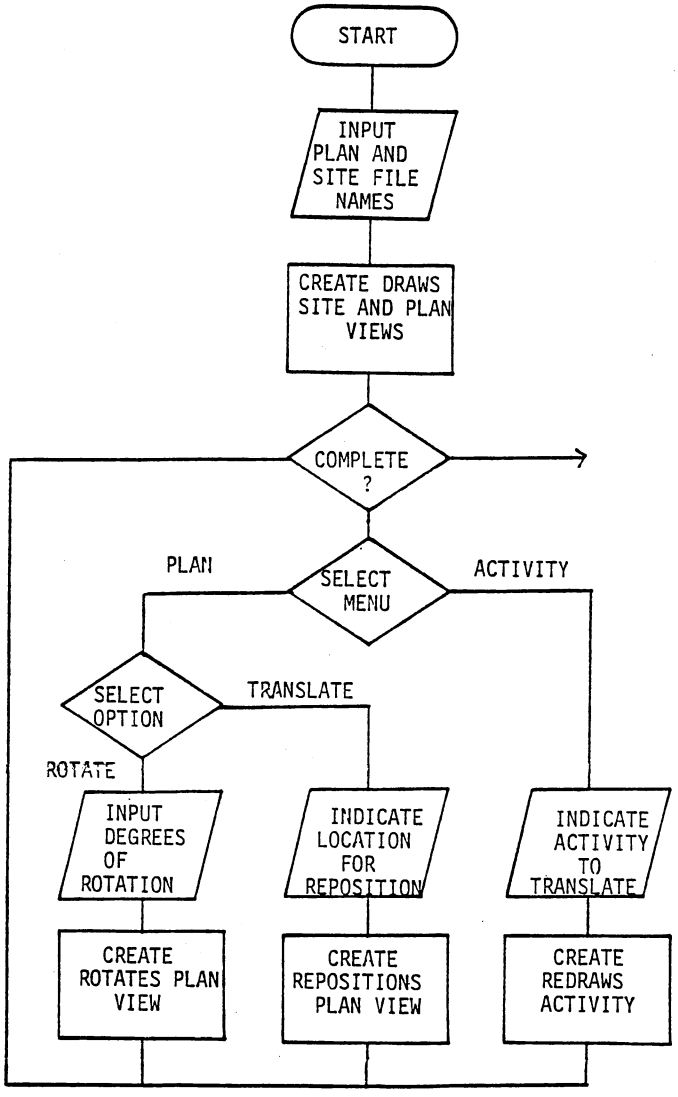


Figure 26. PICTURE Routine Flow Diagram

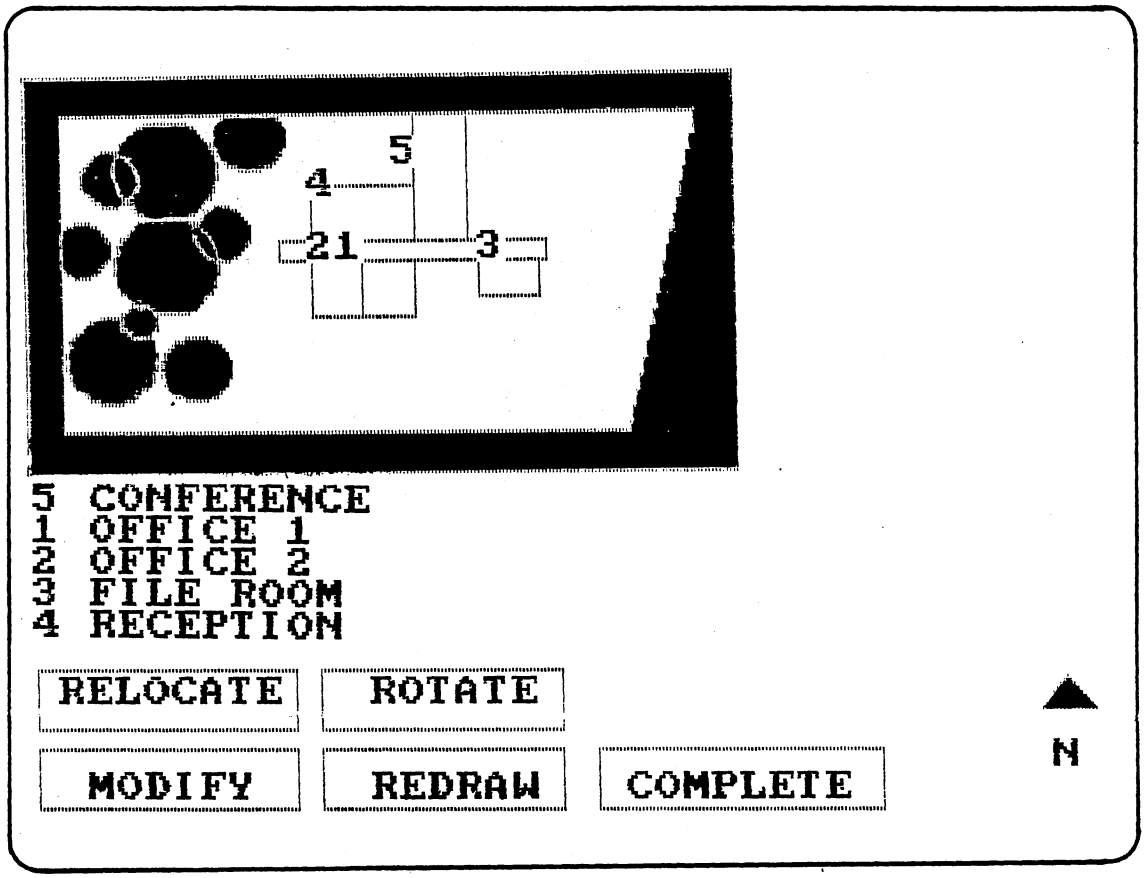


Figure 27. PICTURE Routine Menu Board

The PICTURE routine is used by the designer to fully develop the building concept. He first develops the initial building layout. Next, he develops the site onto which the building layout is placed. He moves and rotates the layout until its orientation is satisfactory. Now, he may call the PICTURE routine and adjust the building layout to integrate better with the site by modifying the position of selected activities. The modified plan can then be placed back on the site. This iterative development process continues until the user is satisfied with the building design. The designer may then call for elevations and modify the building design based on form. Once again he may call PICTURE to check the impact of his modifications on the building plan.

PLANFILE Routine

To the author's knowledge, none of the established computer aided layout techniques allows the user to store a design concept for future reference. CREATE allows both building layouts and site plans to be stored on disk. The plans may be called at a later time, viewed or modified, and restored on a disk.

The PLANFILE routine stores building layouts on File 1 and site plans on File 2. Therefore , a user may compose numerous combinations of building designs and site plans. PLANFILE can be accessed through the four geometry routines -- SPINE, COURTYARD, RADIAL and CLUSTER, through the SITE routine and the PICTURE routine .

Information stored on disk ranges from the type of circulation pattern utilized to the number of trees on the site. Upon storing the information describing a site plan or building layout, PLANFILE always recalls the routine through which the information was passed.

CHAPTER V

SUMMARY AND RECOMMENDATIONS

In summary, this research develops a computer assisted building design technique based on design procedures followed by many designers and taught in the nation's leading schools of architecture. The program, entitled CREATE, is divided into twelve major routines as follows:

1. MAIN Routine - contains the main menu board used to select the desired CREATE option.
2. FUNCTION Routine - used by the designer to establish adjacency requirements between all activities to be included in the building design.
3. SPACE Routine - used by the designer to describe the physical appearance of each building activity.
4. CIRCULATION Routine - contains the menu board used to select the circulation ordering pattern.
5. SPINE Routine - CREATE develops the building layout along a linearly ordered circulation pattern.
6. COURTYARD Routine - CREATE develops the layout around the perimeter of a square courtyard pattern.
7. RADIAL Routine - CREATE develops the plan view using a pinwheel shaped circulation pattern.
8. CLUSTER Routine - user develops a building plan based on own ordering pattern.

9. SITE Routine - used to develop a building site plan including trees and easement setbacks. Building plan is oriented on building site.
10. ELEVATION Routine - used to modify the plan layout based upon building form.
11. PICTURE Routine - auxillary routine used to modify the plan orientation and design.
12. PLANFILE Routine - used to store on disk, site and building plans for future reference.

Computer aided layout techniques have been available since the early sixties, however, they have never realized widespread use. CREATE has a number of advantages over these established techniques. They are as follows :

1. CREATE uses a graphics computer system so the designer may view the design as it is developed. Line drawings are used instead of numbers to indicate activity boundaries. Scale and orientation figures are displayed as necessary.
2. CREATE is user friendly and extremely flexible. In most routines, the user may correct mistakes and modify the design displays. Both keyboard and light pen inputs are allowed. Clearly printed messages and menu boards are displayed across the bottom of the computer screen.
3. Existing computer techniques are programmed around one activity ordering pattern. CREATE utilizes four ordering patterns -- spine, courtyard, radial and cluster. The different patterns allow the designer to quickly develop and evaluate numerous alternative layouts.

4. CREATE displays elevation views of the building layout. The designer may modify the elevation based on a desired building form. These modifications are simultaneously reflected in the plan views.
5. CREATE allows building layout plans and site plans to be stored on disk for future reference and evaluation. In future versions the storage disks may be input to computer aided drafting systems.
6. CREATE allows the designer to develop the building site on which the plan will be built. Often the layout and form design will require modification to better integrate the site environment.
7. CREATE uses a personal computer system. Unlike many existing building layout techniques which require main frame systems, CREATE can be utilized by the average architectural or engineering firm, without investing into a large, expensive system.
8. CREATE is extremely flexible. Future routines such as drafting or perspective views can simply be added to the existing program structure and overlaid into computer memory as required.

CREATE is a first step towards a new generation of computer assisted building design techniques. The computer will never replace the designer, however, it can make his job more productive. Techniques which work alongside the designer, like CREATE, should receive increased

acceptance. This research has developed an initial phase, but the possibilities for expansion are trememdous. Future research areas include:

1. Develop a drafting system to be used in modifying the shapes of both activities and circulation patterns. The system should allow the user to erase and add lines, zoom in and zoom out and add special architectural features such as doorways and windows.
2. Develop a routine to order activities based on form. Activities would be grouped by CREATE, using ceiling heights, proportion, depth or other physical characteristics. The user should be allowed to interact as necessary, as with the existing routines.
3. Develop a routine to develop perspective views of the building plan. The routine should be closely tied to the form ordering routine and the user allowed to modify the perspective view based on form preference.
4. Develop a split screen option so the user may view and modify elevations and plan displays at the same time.
5. Expand the design criteria to include residential and industrial building types.
6. Expand the circulation pattern ordering routines to include multiple patterns within each major ordering scheme. Also, expand the program to allow multiple levels of building plans.
7. Include elevation grade changes in the site development phase.

8. Develop a routine to enable the designer to cut building sections to better study the interior design of the layout.

Computer aided design should once again become a dynamic field of industrial engineering. New concepts, combining practices of architects and engineers, and advances in computer systems should increase the interests and productivity of designers involved with building layout.

A SELECTED BIBLIOGRAPHY

(1) Abercrombie, Stanley. "Workplaces: Assembly Line Architecture." American Institute of Architecture Journal, (July, 1980).

(2) Architect's Journal Handbook. "Analytical Techniques: Layout." (February, 1970).

(3) Buffa, Elwood. "A Heuristic Algorithm and Simulation Approach to Relative Location of Facilities." Management Science, (July, 1963).

(4) Buffa, Elwood. "Allocating Facilities with CRAFT." Harvard Business Review, (March/April, 1964).

(5) Buffa, Elwood, and Thomas E. Vollmann. "The Facilities Layout Problem in Perspective." Management Science, (June, 1966).

(6) Business Week. "CAD Redraws the Architect's Job." (March 15, 1982).

(7) Campion, David. Computers in Architectural Design. London: Elsevier Publishing Company, 1968.

(8) Ching, Francis D. K. Architecture: Form, Space, and Order. New York: Nostrand Reinhold Company, 1979.

(9) Clarke, Andrew. "Computer Aided Drafting--A Critique." Canadian Architect, (February, 1982).

(10) Eastman Charles M. "Mini-Space Planner: A Set of FORTRAN Routines for Building Space-Planning Programs." Spatial Synthesis In Computer Aided Building Design. New York: John Wiley & Sons, 1975.

(11) Eastman, Charles M. Spatial Synthesis in Computer Aided Building Design. New York: John Wiley & Sons, 1975.

(12) Edwards, Harry. "Interactive Minicomputer Graphics in Facilities Planning and Design." Proceedings - AIIE Spring Annual Conference, 1977.

(13) Francis, Richard and John White. Facility Layout and Location. Prentice Hall, 1974.

(14) Gibson, David and Glen Sattoriva. "An Interactive Graphics System for Warehouse Layouts." Proceedings - AIIE Spring Annual Conference, 1978.

(15) Giro, John. Computer Applications in Architecture. London: Applied Science Publishers Ltd., 1978.

(16) Jansan, Jagadish. "A Program for Plotting Plant Layout." Industrial Engineer, (March, 1969).

(17) Johnson, Timothy. IMAGE: An Interactive Graphics Based Computer System for Multi Constrained Spatial Synthesis. MIT: Department of Architecture, 1970.

(18) Laseau, Paul. Graphic Thinking for Architects and Designers. New York: Nostrand Reinhold Company, 1980.

(19) Lee, Kaiman. Computer Aided Architectural Graphics. Boston: Environmental Design and Research Center, 1976.

(20) Lee, Kaiman. Computer Aided Space Planning. Boston: Environmental Design and Research Center, 1976.

(21) Lee, Robert and James Moore. "Corelap - Computerized Relationships Layout Planning." Industrial Engineering, (March, 1967).

(22) Lynch, Kevin. Site Planning. Cambridge: The MIT Press, 1975.

(23) McHarg, Ian L. Design with Nature. New York: American Meseum of Natural History, 1969.

(24) Mitchell, William. Computer Aided Architectural Design New York: Petrocelli Charter, 1977.

(25) Muther, Richard. "Four Approaches To Computerized Layout Planning." Industrial Engineer, (February, 1970).

(26) Muther, Richard. Systematic Layout Planning. 2nd Ed. Boston: CBI Publishing Company, 1973.

(27) Pena, William. Problem Seeking. New York: Cahners Book International, 1971.

(28) Pfekkerkorn, Charles E. "The Design Problem Solver: A System for Designing Equipment or Furniture Layouts." Spatial Synthesis In Computer Aided Building Design. New York: John Wiley & Sons, 1975.

(29) Seehof, Jerrold M. "Automated Layout Design Program." Industrial Engineer, (December, 1967).

(30) Siegel, A. I. "A New Method For the Scientific Layout of Workspaces." Applied Ergonomics, (June, 1982).

(31) Stewart, Clifford. Computer Architecture Programs - 1. Boston: Center for Environmental Research, 1970.

(32) Stewart, Clifford. Computer Architecture Programs - 2. Boston: Center for Environmental Research, 1970.

(33) Stewart, Clifford. Computer Architecture Programs - 3. Boston: Center for Environmental Research, 1970.

(34) Teicholz, E. L. "The Biography of a Remarkable Tool." American Institute of Architecture Journal, (May, 1980).

(35) Tompkins, James. Computer Aided Layout: A User's Guide. AIIE, 1978.

(36) Vollman, Thomas. "A Computerized Model for Office Layout." Industrial Engineer, (July, 1968).

(37) Vollman, Thomas. "An Experimental Comparison of Techniques for the Assignment of Facilities to Locations." Management Science, (January, 1967).

(38) Weinzapfel, Guy. "IMAGE: Computer Assistant for Architectural Design." Spatial Synthesis in Computer Aided Building Design. New York: John Wiley & Sons, 1975.

(39) White, Edward T. Concept Source Book: A Vocabulary of Architectural Forms. Tuscon: Architectural Media, Ltd., 1975.

(40) White, Edward. T. Ordering Systems - An Introduction to Architectural Design. Tuscon: Architectural Media, Ltd., 1973.

APPENDIXES

APPENDIX A

COMPUTER LISTING OF TECHNIQUE

```
10 '
20 ' This is an interactive design technique.  It models the process followed
30 ' by many architectural designers.  The program is organized into twelve
40 ' major routines--main, function, space, geometry, spine, courtyard, radial,
50 ' cluster, site, elevation, picture, and planfile.
60 '*****************************************************************************
70 'MAIN Routine
80 '*****************************************************************************
90 'The major varibles used in this program are:
100 '    TITLE$ -Array containing the labels of each activity as well as the
    '            user defined relationships between activities(alpha).
110 '    TITLE  -Array containing the numeric scores of relationships between
120 '            each activity.
130 '    SPACE  -Array containing information about each activity as follows:
140 '                      1-area in square feet
150 '                      2-length proportion
160 '                      3-width proportion
170 '                      4-ceiling height
180 '                      5-floor level
190 '                      6-wall assignment
200 '                      7-access space
210 '                      8-outside wall requirements
220 '                      9-outside entrance requirements
230 '                     10-modification flag
240 '                     11-number of plan points
250 '    PLANPTS-Array of points describing activiity floor plans
260 '                  1 -Identifies activity
270 '                  2 -Identifies point number
280 '                  3 -Identiifes header, xcoordinate, ycoordinate or pointer
290 '    TEMPPTS  -Array of points used to describe side elevation
300 '    TSPACE   -Total square footage of building
310 '    SCALE    -User defined size of views
320 '    SIDE     -Side dimension of activity floor plan
330 '    LENGTH   -Length dimension of activity floor plan
340 '    BASE     -Baseline of elevation views
350 '    SFLAG    -Indicates type of activity placement
360 '    MAXAREA  -Maximum area of unassigned activities
370 '    WSPINE   -Width of spine area
380 '    LSPINE   -Length of spine area
390 '    CORNER   -Array indicating if reserved corner position occupied
400 '    SPINE    -Array containing coordinates of spine corner points
410 '    YDIST    -Translate distance in the y-direction
420 '    XDIST    -Translate distance in the x-direction
430 '    TRACK    -Array containing corner point coordinates adjacent to pattern
440 '    ACTIVITY -Indicates the activity which is scheduled to be placed
450 '    PLACED   -Indicates the activity which has previously been placed
460 '    NPLACED  -Number of activities placed on pattern
470 '    XPLACE   -X-coordinate of label position
480 '    YPLACE   -Y-coordinate of label position
490 '    IFLAG    -Indicates interactive mode
500 '    MAXREL   -Maximum relationship score between placed and candidates
510 '    MINREL   -Minimum relationship score between placed and candidates
520 '    SFLAG    -Indicates placement scheme selection
530 '    MAXAREA  -Maximum square footage area of candidate activities
```

```
540 '    JUMP      -Indicates ordering scheme selection
550 '    YARD      -Dimension of courtyard
560 '    YARDPTS   -Array containing corner point coordinates of courtyard
570 '    CENTER    -Courtyard dimension
580 '    WHALL     -Width of hallway
590 '    LHALL     -Length of wing hallway
600 '    PINPTS    -Array containing inner courtyard corner point coodinates
610 '    HALLPTS   -Array containing coordinates of all pinwheel corner points
620 '    GRID      -Grid dimension
630 '    SITE      -Array containing site corner point coordinates
640 '    NUMSITEPOINTS-Indicates number of site corner points
650 '    EASE      -Array containing easement setback corner point coordinates
660 '    NUMEASEPOINTS-Indicates number of easement setback corner points
670 '    TREE      -Array containing coordinates of tree locations and tree size
680 '    NUMTREEPOINTS-Indicates number of trees located on site
690 '    DEGREE    -Indicates the desired number of degrees of rotation
700 '    ELEV$     -Indicates elevation view
710 '
720 KEY OFF
730 LPRINT CHR$(27) CHR$(71)
740 LPRINT CHR$(27)"C"CHR$(66);
750 LPRINT CHR$(27)"N"CHR$(6);
760 DIM TITLE$(20,20),TITLE(20,20),SPACE(20,11),PLANPTS(20,10,4),
    TEMPPTS(10,4),TRACK(20,9),CORNER(4),YARDPTS(4,2),PINPTS(4,2),HALLPTS(12,2)
770 DIM SITE(10,2),EASE(10,2),SPINE(4,2),TREE(10,3),TEMP(20,1)
780 SCREEN 1,0
790 COLOR 1,1
800 CLS:SCALE=1
810 LOCATE 8,16:PRINT "CREATE:"
820 LOCATE 10,1:PRINT "INTERACTIVE MODEL OF THE DESIGN PROCESS"
830 LOCATE 23,1:PRINT " CREATE BLDG CREATE PLAN  MODIFY PLAN "
840 LINE (2,185)-(97,170),1,B: LINE (100,185)-(196,170),1,B:
    LINE (198,185)-(300,170),1,B
850 PEN ON
860 IF PEN(3)=0 THEN GOTO 850
870 IF PEN(9)<11 THEN GOTO 920
880 IF PEN(9)>=11 AND PEN(9)<23 THEN GOTO 910
890 IF PEN(9)>=23 THEN GOTO 900 ELSE GOTO STOP
900 CHAIN MERGE "PICTURE",10,ALL,DELETE 10-890
910 CHAIN MERGE "SITE",10,ALL,DELETE 10-900
920 CHAIN MERGE "FUNCTION",10,ALL,DELETE 10-910
```

```
10 CLS
20 '*****************************************************************************
30 'FUNCTION Routine
40 '*****************************************************************************
50 'This routine collects the activity titles and information describing
60 'the relationships between each of these activities.
70 '
80 CLS
90 LOCATE 5,1: PRINT SPC(11) "********************"
100 LOCATE 6,12:PRINT          "* FUNCTION Routine *"
110 LOCATE 7,1: PRINT SPC(11) "********************"
120 FOR A=1 TO 500
130 NEXT A
140 CLS
150 LOCATE 1,12:PRINT "FUNCTION Routine"
160 LOCATE 3,1:INPUT "ENTER NUMBER OF ACTIVITIES   ",N
170 LOCATE 5,1:PRINT "ENTER TITLES OF ACTIVITIES"
180 FOR A=1 TO N
190     LOCATE 5+A,2:PRINT A:LOCATE 5+A,5:INPUT " ",TITLE$(A,A)
200 NEXT A
210 LOCATE 23,1:PRINT "ANY CHANGES?        YES   NO "
220 LINE (150,185)-(175,170),1,B:LINE (195,185)-(220,170),1,B
230 PEN ON
240 IF PEN(3)=0 THEN GOTO 230
250 IF PEN(8)>21 AND PEN(9)>19 AND PEN(9)<23 THEN GOTO 80
260 '
270 'Enter activity relationship information and calculate scores
280 '
290 CLS
300 LOCATE 1,12:PRINT "FUNCTION Routine"
310 LOCATE 7,5: PRINT "SEPARATION NECESSARY?    NO    YES"
320 LOCATE 9,5: PRINT "SHARED PERSONNEL"
330 LOCATE 10,5:PRINT "SHARED EQUIPMENT"
340 LOCATE 11,5:PRINT "COMMON FUNCTIONS"
350 LOCATE 12,5:PRINT "VISUAL ACCESS REQUIREMENTS"
360 LOCATE 13,5:PRINT "SECURITY REQUIREMENTS"
370 LOCATE 14,5:PRINT "SUPERVISORY REQUIREMENTS"
380 LOCATE 15,5:PRINT "PERSONNEL CONVENIENCE"
390 LOCATE 17,5:PRINT "MATERIAL FLOW"
400 LOCATE 18,5:PRINT "PEOPLE FLOW"
410 LOCATE 22,1:PRINT "     NONE     LOW     MEDIUM    HIGH "
420 LINE(225,45)-(255,60),1,B:LINE(275,45)-(305,60),1,B
430 LINE(28,160)-(83,180),1,B:LINE(98,160)-(153,180),1,B:LINE(170,160)-(225,180
    ),1,B:LINE(240,160)-(295,180),1,B
440 FOR A=1 TO N-1
450     LOCATE 3,1:PRINT SPC(14)
460     LOCATE 3,1:PRINT TITLE$(A,A)
470     LOCATE 3,15: PRINT "vs"
480     FOR B=A+1 TO N
490         LOCATE 3,20:PRINT SPC(20)
500         LOCATE 3,20:PRINT TITLE$(B,B)
510         FLAG1=0
520         FOR C=7 TO 18
530             IF C=8 OR C=16 THEN C=C+1
540             LOCATE C,3:PRINT "*"
550             PEN ON
560             IF PEN(3)=0 THEN GOTO 550
570             IF FLAG1=1 THEN GOTO 590
580             IF FLAG1=0 AND PEN(9)>32 THEN TITLE$(A,B)="X":TITLE$(B,A)="X":
                  TITLE(A,B)=-30:TITLE(B,A)=-30:GOTO 690 ELSE GOTO 630
```

```
590             IF PEN(9)>=5 AND PEN(9)<=10 THEN TITLE(A,B)=TITLE(A,B)+0:
                 TITLE(B,A)=TITLE(B,A)+0:GOTO 640 ELSE GOTO 600
600             IF PEN(9)>=14 AND PEN(9)<=18 THEN TITLE(A,B)=TITLE(A,B)+1:
                 TITLE(B,A)=TITLE(B,A)+1:GOTO 640 ELSE GOTO 610
610             IF PEN(9)>=22 AND PEN(9)<=29 THEN TITLE(A,B)=TITLE(A,B)+2:
                 TITLE(B,A)=TITLE(B,A)+2:GOTO 640 ELSE GOTO 620
620             IF PEN(9)>=32 AND PEN(9)<=37 THEN TITLE(A,B)=TITLE(A,B)+3:
                 TITLE(B,A)=TITLE(B,A)+3:GOTO 640 ELSE STOP
630             FLAG1=1
640             PEN OFF
650             LOCATE C,3:PRINT SPC(1)
660             FOR D=1 TO 500
670             NEXT D
680         NEXT C
690     NEXT B
700 NEXT A
710 CLS
720 '
730 'Assign activity relationship measures
740 '
750 FOR A=1 TO N-1
760     FOR B=A+1 TO N
770         IF TITLE(A,B)>=22 THEN TITLE$(A,B)="A":TITLE$(B,A)="A":GOTO 800
            ELSE GOTO 780
780         IF TITLE(A,B)>=17 THEN TITLE$(A,B)="B":TITLE$(B,A)="B":GOTO 800
            ELSE GOTO 790
790         IF TITLE(A,B)>=0  THEN TITLE$(A,B)="C":TITLE$(B,A)="C"
             ELSE GOTO 800
800     NEXT B
810 NEXT A
820 '
830 'Display relationship measures and allow modifications
840 '
850 CLS:C=0
860 LOCATE 2,8:PRINT "FUNCTIONAL RELATIONSHIPS"
870 LOCATE 4,1:PRINT"ACTIVITY  VS  ACTIVITY      RATING"
880 FOR A=1 TO N-1
890     FOR B=A+1 TO N
900         C=C+1
910         IF C>14 THEN C=1:FLAG2=1:GOTO 990 ELSE GOTO 920
920         LOCATE C+6,1:PRINT SPC(39):LOCATE C+6,1:PRINT TITLE$(A,A),
             TITLE$(B,B);
930         LOCATE C+6,32:PRINT TITLE$(A,B)
940     NEXT B
950 NEXT A
960 FOR A=C+7 TO 21
970     PRINT SPC(39)
980 NEXT A
990 LOCATE 23,1:PRINT "ANY CHANGES?        YES   NO "
1000 LINE (150,185)-(175,170),1,B:LINE (195,185)-(220,170),1,B
1010 PEN ON
1020 IF PEN(3)=0 THEN GOTO 1010
1030 IF PEN(8)>21 AND PEN(9)>24 AND FLAG2=0 THEN GOTO 1170
      ELSE IF PEN(8)>21 AND PEN(9)>24 AND FLAG2=1 THEN FLAG2=0:GOTO 920
       ELSE GOTO 1040
1040 LINE (150,185)-(175,170),0,B:LINE (195,185)-(220,170),0,B
1050 LOCATE 23,1:INPUT "ENTER ACTIVITIES AND RELATIONSHIP        ",ACTIVITY1$,
     ACTIVITY2$,NEWREL$
1060 FOR A=1 TO N-1
1070     FOR B=A+1 TO N
```

```
1080          IF ACTIVITY1$=TITLE$(A,A) AND ACTIVITY2$=TITLE$(B,B) THEN
               TITLE$(A,B)=NEWREL$ ELSE GOTO 1140
1090          IF NEWREL$="A" THEN TITLE(A,B)=27:TITLE(B,A)=27 ELSE GOTO 1100
1100          IF NEWREL$="B" THEN TITLE(A,B)=21:TITLE(B,A)=21 ELSE GOTO 1110
1110          IF NEWREL$="C" THEN TITLE(A,B)=17:TITLE(B,A)=17 ELSE GOTO 1120
1120          IF NEWREL$="X" THEN TITLE(A,B)=-30:TITLE(B,A)=-30 ELSE GOTO 1130
1130          GOTO 850
1140      NEXT B
1150 NEXT A
1160 STOP
1170 CHAIN MERGE "SPACE",10,ALL,DELETE 10-1160
```

```
10 '*************************************************************************
20 'SPACE Routine
30 '*************************************************************************
40 'This routine collects information concerning the space required by
50 'each activity.  The user enters parameters such as area, ceiling height
60 'and proportion and the computer draws a plan and a section view.
70 '
80 CLS
90 LOCATE 5,1: PRINT SPC(12) "*****************"
100 LOCATE 6,13:PRINT          "* SPACE Routine *"
110 LOCATE 7,1: PRINT SPC(12) "*****************"
120 FOR B=1 TO 500
130 NEXT B
140 CLS:TSPACE=0:SCALE=1:TEMP=0
150 LOCATE 1,12:PRINT "SPACE Routine"
160 'IF JUMP=1 THEN GOTO 180
170 FOR A=1 TO N
180     ' JUMP=0
190     LOCATE 3,1: PRINT TITLE$(A,A)
200     LOCATE 5,1: PRINT "AREA  "
210     LOCATE 6,1: PRINT "PROPORTION  "
220     LOCATE 7,1: PRINT "CEILING HEIGHT  "
230     LOCATE 8,1: PRINT "FLOOR LEVEL  "
240     LOCATE 10,1:PRINT "WALL ASSIGNMENT  "
250     LOCATE 11,1:PRINT "ADJACENCY REQ'D  "
260     'LOCATE 15,1:PRINT "OUTSIDE WINDOW?    YES   NO "
270     'LOCATE 18,1:PRINT "OUTSIDE ENTRANCE?  YES   NO "
280     'LINE(150,120)-(175,105),1,B:LINE(195,120)-(220,105),1,B
290     'LINE(150,145)-(175,130),1,B:LINE(195,145)-(220,130),1,B
300     LOCATE 5,20: INPUT " ",SPACE(A,1)
310     TSPACE=TSPACE+SPACE(A,1)
320     LOCATE 6,20: INPUT " ",SPACE(A,2):LOCATE 6,22:INPUT " TO ",SPACE(A,3)
330     IF SPACE(A,2)=0 AND SPACE(A,3)=0 THEN SPACE(A,2)=1:SPACE(A,3)=1:
         LOCATE 6,20:PRINT 1:LOCATE 6,26:PRINT 1 ELSE GOTO 340
340     LOCATE 7,20: INPUT " ",SPACE(A,4)
350     IF SPACE(A,4)=0 THEN SPACE(A,4)=10:LOCATE 7,20:PRINT 10 ELSE GOTO 360
360     LOCATE 8,20: INPUT " ",SPACE(A,5)
370     IF SPACE(A,5)=0 THEN SPACE(A,5)=1:LOCATE 8,20:PRINT 1 ELSE GOTO 380
380     LOCATE 10,20:INPUT " ",TEMP1$
390     LOCATE 11,20:INPUT " ",TEMP2$
400     IF TEMP1$="NORTH" THEN SPACE(A,6)=1:GOTO 450 ELSE GOTO 410
410     IF TEMP1$="EAST" THEN SPACE(A,6)=2: GOTO 450 ELSE GOTO 420
420     IF TEMP1$="SOUTH" THEN SPACE(A,6)=3:GOTO 450 ELSE GOTO 430
430     IF TEMP1$="WEST" THEN SPACE(A,6)=4: GOTO 450 ELSE GOTO 440
440     SPACE(A,6)=0
450     FOR B=1 TO N
460         IF TEMP2$=TITLE$(B,B) THEN TITLE$(A,B)="A":TITLE$(B,A)="A":
             TITLE(B,A)=99:TITLE(A,B)=99 ELSE GOTO 470
470     NEXT B
480     'PEN ON
490     'IF PEN(3)=0 THEN GOTO 470
500     'IF PEN(9)>23 THEN SPACE(A,8)=0 ELSE SPACE(A,8)=1
510     'PEN OFF
520     'FOR B=1 TO 500
530     'NEXT B
540     'PEN ON
550     'IF PEN(3)=0 THEN GOTO 530
560     'IF PEN(9)>23 THEN SPACE(A,9)=0 ELSE SPACE(A,9)=1
570     'PEN OFF
580     CLS
```

```
590 '
600 'Calculate length and side dimensions
610 '
620     SIDE=SQR(SPACE(A,1)*SPACE(A,3)/SPACE(A,2))*SCALE
630     LENGTH=SIDE*SPACE(A,2)/SPACE(A,3)
640     FOR B=1 TO 4
650         PLANPTS(A,B,1)=B
660         PLANPTS(A,B,4)=B+1
670         IF B=4 THEN PLANPTS(A,B,4)=1
680     NEXT B
690     SPACE(A,11)=4
700     PLANPTS(A,1,2)=50        :PLANPTS(A,1,3)=80
710     PLANPTS(A,2,2)=50+LENGTH:PLANPTS(A,2,3)=80
720     PLANPTS(A,3,2)=50+LENGTH:PLANPTS(A,3,3)=80+SIDE
730     PLANPTS(A,4,2)=50        :PLANPTS(A,4,3)=80+SIDE
740     PI=3.141593:ANGLE=.5*PI
750     FOR B=1 TO SPACE(A,11)
760         IF B=4 THEN GOTO 810
770         TEMPPTS(B,1)=PLANPTS(A,B,1)
780         TEMPPTS(B,2)=PLANPTS(A,B,2)*COS(ANGLE)-PLANPTS(A,B,3)*SIN(ANGLE)
             +PLANPTS(A,4,2)*(1-COS(ANGLE))+PLANPTS(A,4,3)*SIN(ANGLE)
790         TEMPPTS(B,3)=PLANPTS(A,B,2)*SIN(ANGLE)+PLANPTS(A,B,3)*COS(ANGLE)
             +PLANPTS(A,4,3)*(1-COS(ANGLE))-PLANPTS(A,4,2)*SIN(ANGLE)
800         TEMPPTS(B,4)=PLANPTS(A,B,4)
810     NEXT B
820     TEMPPTS(4,1)=PLANPTS(A,4,1):TEMPPTS(4,4)=PLANPTS(A,4,4):
830     TEMPPTS(4,2)=PLANPTS(A,4,2):TEMPPTS(4,3)=PLANPTS(A,4,3)
840     FOR B=1 TO SPACE(A,11)
850         TEMPPTS(B,2)=TEMPPTS(B,2)+ABS(PLANPTS(A,2,2)-PLANPTS(A,1,2))+35
860     NEXT B
870 '
880 'Display plan view, front elevation and side elevation
890 '
900     IF SPACE(A,6)=0 THEN WALL$="      "
910     IF SPACE(A,6)=1 THEN WALL$="NORTH"
920     IF SPACE(A,6)=2 THEN WALL$="EAST"
930     IF SPACE(A,6)=3 THEN WALL$="SOUTH"
940     IF SPACE(A,6)=4 THEN WALL$="WEST"
950     CLS:LOCATE 19,1:PRINT TITLE$(A,A):LOCATE 20,1:PRINT "WALL ASSIGNMENT: ";
         WALL$
960     BASE=PLANPTS(A,1,3)-35
970     FLAG=1:I=1:COUNT=0
980     WHILE FLAG
990        I=PLANPTS(A,I,1)
1000        LINE (PLANPTS(A,PLANPTS(A,I,1),2),PLANPTS(A,PLANPTS(A,I,1),3))-
             (PLANPTS(A,PLANPTS(A,I,4),2),PLANPTS(A,PLANPTS(A,I,4),3)),1
1010        LINE (PLANPTS(A,I,2),BASE)-(PLANPTS(A,PLANPTS(A,I,4),2),BASE),1
1020        LINE (PLANPTS(A,I,2),BASE-SPACE(A,4)*SCALE)-
                  (PLANPTS(A,PLANPTS(A,I,4),2),BASE-SPACE(A,4)*SCALE),1
1030        LINE (PLANPTS(A,I,2),BASE)-(PLANPTS(A,I,2),BASE-SPACE(A,4)*SCALE),1
1040        LINE (TEMPPTS(I,2),BASE)-(TEMPPTS(TEMPPTS(I,4),2),BASE),1
1050        LINE (TEMPPTS(I,2),BASE-SPACE(A,4)*SCALE)-
                 (TEMPPTS(TEMPPTS(I,4),2),BASE-SPACE(A,4)*SCALE),1
1060        LINE (TEMPPTS(I,2),BASE)-(TEMPPTS(I,2),BASE-SPACE(A,4)*SCALE),1
1070        I=PLANPTS(A,I,4)
1080        COUNT=COUNT+1
1090        IF COUNT>=SPACE(A,11) THEN FLAG=0
1100    WEND
1110 '
1120 'Display scale figure
```

```
1130 '
1140    LINE (10,BASE)-(11,BASE-2*SCALE),1
1150    LINE (12,BASE)-(11,BASE-2*SCALE),1
1160    CIRCLE (11,BASE-5*SCALE),1
1170    CIRCLE (11,BASE-3*SCALE),1*SCALE
1180    IF A<TEMP THEN A=A+1:GOTO 740 ELSE GOTO 1190
1190 '
1200 'Allow modifications to be made to activity shape
1210 '
1220    LOCATE 23,1:PRINT "ANY CHANGES?       YES   NO "
1230    LINE (150,185)-(175,170),1,B:LINE (195,185)-(220,170),1,B
1240    PEN ON
1250    IF PEN(3)=0 THEN GOTO 1240
1260    IF PEN(8)>21 AND PEN(9)>24 THEN GOTO 2040
1270    LINE (150,185)-(175,170),0,B:LINE (195,185)-(220,170),0,B
1280    LOCATE 23,1:PRINT SPC(39)
1290    LOCATE 23,1:PRINT " MODIFY REDRAW RESCALE COMPLETE FORMAT"
1300    LINE(2,185)-(60,170),1,B:LINE(62,185)-(116,170),1,B:LINE(118,185)-
          (179,170),1,B:LINE(183,185)-(248,170),1,B:LINE(251,185)-(318,170),1,B
1310    PEN ON
1320    IF PEN(3)=0 THEN GOTO 1310
1330    IF PEN(9)<=7 THEN GOTO 1380
1340    IF PEN(9)>7 AND PEN(9)<=14 THEN GOTO 740
1350    IF PEN(9)>14 AND PEN(9)<=22 THEN GOTO 1870
1360    IF PEN(9)>22 AND PEN(9)<=32 THEN GOTO 2040
1370    IF PEN(9)>32 THEN JUMP=1:GOTO 140 ELSE STOP
1380 '
1390 'Allow modifications to be made to the building plan
1400 '
1410    LINE(2,185)-(60,170),0,B:LINE(62,185)-(116,170),0,B:LINE(118,185)-
          (179,170),0,B:LINE(183,185)-(248,170),0,B:LINE(251,185)-(318,170),0,B
1420    LINE (150,185)-(175,170),0,B:LINE (195,185)-(220,170),0,B
1430    LOCATE 23,1:PRINT SPC(39):SPACE(A,10)=1
1440    LOCATE 21,1:PRINT "INDICATE NEW WALL ORIENTATION"
1450    PEN ON
1460    IF PEN(3)=0 THEN GOTO 1450
1470    X1=PEN(4):Y1=PEN(5)
1480    PSET (X1,Y1)
1490    FOR B=1 TO 500:NEXT B
1500    LOCATE 23,1:PRINT "CORRECT POINT?     YES   NO       "
1510    LINE (150,185)-(175,170),1,B:LINE (195,185)-(220,170),1,B
1520    PEN ON
1530    IF PEN(3)=0 THEN GOTO 1520
1540    IF PEN(8)>21 AND PEN(9)>24 THEN PSET (X1,Y1),0:GOTO 1440 ELSE GOTO 1550
1550    PSET (X1,Y1),0
1560    FOR B=1 TO 500:NEXT B
1570    PEN ON
1580    IF PEN(3)=0 THEN GOTO 1570
1590    X2=PEN(4):Y2=PEN(5)
1600    PSET (X2,Y2)
1610    FOR B=1 TO 500:NEXT B
1620    PEN ON
1630    IF PEN(3)=0 THEN GOTO 1620
1640    IF PEN(8)>21 AND PEN(9)>24 THEN PSET (X2,Y2),0:GOTO 1560 ELSE GOTO 1650
1650    PSET (X2,Y2),0
1660    FOR B=1 TO 500:NEXT B
1670    SPACE(A,11)=SPACE(A,11)+1
1680    PLANPTS(A,SPACE(A,11),2)=X2:PLANPTS(A,SPACE(A,11),3)=Y2:
          PLANPTS(A,SPACE(A,11),1)=SPACE(A,11)
1690    FOR B=1 TO SPACE(A,11)
```

```
1700        IF X1>=PLANPTS(A,B,2)-5 AND X1<=PLANPTS(A,B,2)+5 AND
          Y1>=PLANPTS(A,B,3)-5 AND Y1<=PLANPTS(A,B,3)+5 THEN GOTO 1730
1710    NEXT B
1720    STOP
1730    PLANPTS(A,SPACE(A,11),4)=PLANPTS(A,B,4)
1740    PLANPTS(A,B,4)=SPACE(A,11)
1750    LINE(PLANPTS(A,B,2),PLANPTS(A,B,3))-(PLANPTS(A,SPACE(A,11),2),
          PLANPTS(A,SPACE(A,11),3)),1
1760    LOCATE 21,1:PRINT SPC(39)
1770    LOCATE 23,1:PRINT SPC(39)
1780    LOCATE 23,1:PRINT "ANOTHER WALL?       YES   NO"
1790    LINE (150,185)-(175,170),1,B:LINE (195,185)-(220,170),1,B
1800    PEN ON
1810    IF PEN(3)=0 THEN GOTO 1800
1820    IF PEN(8)>21 AND PEN(9)>24 THEN GOTO 1270
1830    GOTO 1410
1840 '
1850 ' Rescale building views
1860 '
1870    LINE(2,185)-(60,170),0,B:LINE(62,185)-(116,170),0,B:LINE(118,185)-
          (179,170),0,B:LINE(183,185)-(248,170),0,B:LINE(251,185)-(318,170),0,B
1880    LOCATE 23,1:PRINT SPC(39)
1890    LOCATE 23,1:PRINT "SCALE CURRENTLY IS ":LOCATE 23,20:PRINT SCALE
1900    LOCATE 24,1:INPUT "ENTER DESIRED SCALE ",SCALE
1910    FOR B=1 TO A
1930       FOR C=1 TO SPACE(B,11)
1940          PLANPTS(B,C,2)=PLANPTS(B,C,2)*SCALE
1950          PLANPTS(B,C,3)=PLANPTS(B,C,3)*SCALE
1960       NEXT C
1970       XDIST=50-PLANPTS(B,1,2):YDIST=80-PLANPTS(B,1,3)
1980       FOR C=1 TO SPACE(B,11)
1990          PLANPTS(B,C,2)=PLANPTS(B,C,2)+XDIST
2000          PLANPTS(B,C,3)=PLANPTS(B,C,3)+YDIST
2010       NEXT C
2020    NEXT B
2030    TEMP=A:A=1:GOTO 740
2040 '
2050 ' Draw window and doors as required
2060 '
2070    'IF SPACE(A,8)=0 THEN GOTO 2040
2080    'LINE (PLANPTS(A,1,2)+LENGTH/4,BASE-4)-(PLANPTS(A,1,2)+LENGTH/4+3*SCALE,
              BASE-6*SCALE),1,B
2090    'IF SPACE(A,9)=0 THEN GOTO 2060
2100    'LINE (PLANPTS(A,1,2)+LENGTH/2,BASE)-(PLANPTS(A,1,2)+LENGTH/2+3*SCALE,
            BASE-6*SCALE),1,B
2110    FOR B=1 TO 1000:NEXT B:CLS
2120 NEXT A
2130 CHAIN MERGE "GEOMTRY",10,ALL,DELETE 10-2120
```

```
10 '*************************************************************************
20 ' CIRCULATION Routine
30 '*************************************************************************
40 ' This routine contains the menu board to select then desired circulation
50 ' ordering scheme.  Schemes include spine, courtyard, radial and cluster.
60 '
70 CLS:PEN OFF
80  LOCATE 5,1:PRINT SPC(11) "***********************
90  LOCATE 6,12:PRINT        "* CIRCULATION Routine *
100 LOCATE 7,1:PRINT SPC(11) "***********************
110 FOR A=1 TO 4
120    CORNER(A)=0
130 NEXT A
140 FOR A=1 TO 500:NEXT A
150 CLS
160 LOCATE 2,13:PRINT "GEOMETRY Routine"
170 LOCATE 25,1:PRINT "  INDICATE DESIRED PLACEMENT SCHEME"
180 LOCATE 23,1:PRINT "  SPINE    COURTYD  RADIAL   CLUSTER"
190 LINE(5,185)-(75,170),1,B:LINE(80,185)-(150,170),1,B:LINE(155,185)-(225,170)
    ,1,B:LINE(230,185)-(299,170),1,B
200 PEN ON
210 IF PEN(3)=0 THEN GOTO 200
220 IF PEN(9)<=10 THEN GOTO 270
230 IF PEN(9)>=11 AND PEN(9)<21 THEN GOTO 280
240 IF PEN(9)>=21 AND PEN(9)<29 THEN GOTO 290
250 IF PEN(9)>=29 THEN GOTO 300
260 STOP
270 CHAIN MERGE "LINE",10,ALL,DELETE 10-260
280 CHAIN MERGE "POINT",10,ALL,DELETE 10-270
290 CHAIN MERGE "RADIAL",10,ALL,DELETE 10-280
300 CHAIN MERGE "CLUSTER",10,ALL,DELETE 10-290
```

```
10 '*************************************************************************
20 'SPINE Routine
30 '*************************************************************************
40 'This routine orders activities in a linear circulation pattern.  The
50 'user inputs the spine width and length.  Plans can be stored on disk.
60 '
70 CLS:PEN OFF
80 FOR A=1 TO 4
90       CORNER(A)=0
100 NEXT A
110 CFLAG=0:JUMP=1
120 LOCATE 2,13:PRINT "LINE Subroutine"
130 LOCATE 8,1  :PRINT "   SIMPLE"
140 LOCATE 14,1 :PRINT "   OFFSET"
150 LINE (10,50)-(80,65),1,B:LINE (10,100)-(80,115),1,B
160 LINE (120,55)-(180,60),1,B
170 LINE (120,105)-(150,110),1,B:LINE (150,110)-(180,115),1,B
180 PEN ON
190 IF PEN(3)=0 THEN GOTO 180
200 IF PEN(8)>=7 AND PEN(8)<=11 AND PEN(9)<=15 THEN GOTO 230
210 IF PEN(8)>11 AND PEN(8)<=15 AND PEN(9)<=15 THEN GOTO 220
220 STOP
230 '
240 'Simple line placement
250 '
260 LOCATE 23,1:INPUT "ENTER SPINE DIMENSIONS  ",LSPINE:
       LOCATE 23,28:INPUT " BY ",WSPINE
270 LSPINE=LSPINE*SCALE:WSPINE=WSPINE*SCALE
280 CLS
290 LOCATE 23,1:PRINT "INTERACTIVE ?      YES   NO"
300 LINE (150,185)-(175,170),1,B:LINE (195,185)-(220,170),1,B
310 PEN ON
320 IF PEN(3)=0 THEN GOTO 310
330 IF PEN(8)>21 AND PEN(9)>24 THEN IFLAG=0 ELSE IFLAG=1
340 CLS
350 '
360 'Draw skeleton spine
370 '
380 RENDPT=(320-LSPINE)/2+LSPINE:LENDPT=(320-LSPINE)/2
390 SPINE(1,1)=LENDPT:SPINE(1,2)=50
400 SPINE(2,1)=RENDPT:SPINE(2,2)=50
410 SPINE(3,1)=RENDPT:SPINE(3,2)=50+WSPINE
420 SPINE(4,1)=LENDPT:SPINE(4,2)=50+WSPINE
430 LINE (SPINE(1,1),SPINE(1,2))-(SPINE(3,1),SPINE(3,2)),1,B
440 '
450 'Determine activity with largest square footage
460 '
470 MAXAREA=0:NPLACED=0
480 FOR A=1 TO N
490    IF SPACE(A,1)>MAXAREA THEN ACTIVITY=A:MAXAREA=SPACE(A,1) ELSE GOTO 500
500 NEXT A
510 SIDE=ABS(PLANPTS(ACTIVITY,2,3)-PLANPTS(ACTIVITY,3,3))
520 LENGTH=ABS(PLANPTS(ACTIVITY,1,2)-PLANPTS(ACTIVITY,2,2))
530 '
540 'Check wall assignment and determine initial activity placement
550 '
560 IF SPACE(ACTIVITY,6)=4 THEN GOTO 620
570 IF SPACE(ACTIVITY,6)=3 THEN GOTO 620
580 IF SPACE(ACTIVITY,6)=2 THEN GOTO 600
590 IF SPACE(ACTIVITY,6)=0 THEN SPACE(ACTIVITY,6)=1
```

```
600 XDIST=160-PLANPTS(ACTIVITY,4,2):YDIST=50-PLANPTS(ACTIVITY,4,3)
610 GOTO 640
620 XDIST=160-PLANPTS(ACTIVITY,2,2):YDIST=50+WSPINE-PLANPTS(ACTIVITY,2,3)
630 GOTO 640
640 '
650 'Translate activity coordinates to spine
660 '
670 FOR A=1 TO SPACE(ACTIVITY,11)
680     PLANPTS(ACTIVITY,A,2)=PLANPTS(ACTIVITY,A,2)+XDIST
690     PLANPTS(ACTIVITY,A,3)=PLANPTS(ACTIVITY,A,3)+YDIST
700 NEXT A
710 '
720 IF PLANPTS(ACTIVITY,3,2)>RENDPT AND PLANPTS(ACTIVITY,3,3)=50 THEN
     CORNER(2)=1
730 IF PLANPTS(ACTIVITY,2,2)>RENDPT AND PLANPTS(ACTIVITY,2,3)=50+WSPINE THEN
     CORNER(3)=1
740 IF PLANPTS(ACTIVITY,4,2)<LENDPT AND PLANPTS(ACTIVITY,4,3)=50 THEN
     CORNER(1)=1
750 IF PLANPTS(ACTIVITY,1,2)<LENDPT AND PLANPTS(ACTIVITY,1,3)=50+WSPINE THEN
     CORNER(4)=1
760 '
770 'Draw activity on spine
780 '
790   FLAG=1:I=1:COUNT=0
800   IF CFLAG=0 THEN NPLACED=NPLACED+1
810    WHILE FLAG
820       I=PLANPTS(ACTIVITY,I,1)
830       LINE (PLANPTS(ACTIVITY,PLANPTS(ACTIVITY,I,1),2),PLANPTS(ACTIVITY,
           PLANPTS(ACTIVITY,I,1),3))-(PLANPTS(ACTIVITY,PLANPTS(ACTIVITY,I,4),2)
           ,PLANPTS(ACTIVITY,PLANPTS(ACTIVITY,I,4),3)),1
840       I=PLANPTS(ACTIVITY,I,4)
850       COUNT=COUNT+1
860       IF COUNT>=SPACE(ACTIVITY,11) THEN FLAG=0
870    WEND
880 '
890 'Store reference points located on spine
900 '
910 IF CFLAG=0 THEN GOTO 960
920 FOR A=1 TO NPLACED
930     IF ACTIVITY=TRACK(A,1) THEN M=A:GOTO 980 ELSE GOTO 940
940 NEXT A
950 STOP
960 M=NPLACED
970 TRACK(M,1)=ACTIVITY
980 IF PLANPTS(ACTIVITY,4,3)=50 THEN GOTO 990 ELSE GOTO 1030
990 TRACK(M,2)=0                        :TRACK(M,3)=0
1000 TRACK(M,4)=0                        :TRACK(M,5)=0
1010 TRACK(M,6)=PLANPTS(ACTIVITY,3,2):TRACK(M,7)=PLANPTS(ACTIVITY,3,3)
1020 TRACK(M,8)=PLANPTS(ACTIVITY,4,2):TRACK(M,9)=PLANPTS(ACTIVITY,4,3):GOTO 1080

1030 IF PLANPTS(ACTIVITY,1,3)=50+WSPINE THEN GOTO 1040 ELSE STOP
1040 TRACK(M,2)=PLANPTS(ACTIVITY,1,2):TRACK(M,3)=PLANPTS(ACTIVITY,1,3)
1050 TRACK(M,4)=PLANPTS(ACTIVITY,2,2):TRACK(M,5)=PLANPTS(ACTIVITY,2,3)
1060 TRACK(M,6)=0                        :TRACK(M,7)=0
1070 TRACK(M,8)=0                        :TRACK(M,9)=0:GOTO 1080
1080 '
1090 'Label floor plan
1100 '
1110 XPLACE=INT(PLANPTS(ACTIVITY,1,2)/8):YPLACE=INT(PLANPTS(ACTIVITY,1,3)/8)
1120 IF TRACK(M,7)=50 THEN LOCATE YPLACE,XPLACE+1:GOTO 1140 ELSE GOTO 1130
```

```
1130 IF TRACK(M,3)=50+WSPINE THEN LOCATE YPLACE+5,XPLACE+1:GOTO 1140 ELSE STOP
1140 IF ACTIVITY< 10 THEN PRINT USING "#"; ACTIVITY
1150 IF ACTIVITY>=10 THEN PRINT USING "##";ACTIVITY
1160 IF CFLAG=1 THEN GOTO 1280
1170 IF NPLACED<= 6 THEN GOTO 1200
1180 IF NPLACED<=12 THEN GOTO 1230
1190 IF NPLACED<=20 THEN GOTO 1260
1200 LOCATE 12+NPLACED,1:PRINT USING "#";ACTIVITY
1210 LOCATE 12+NPLACED,3:PRINT USING "\         \";TITLE$(ACTIVITY,ACTIVITY)
1220 GOTO 1280
1230 LOCATE 6+NPLACED,14:PRINT USING "#";ACTIVITY
1240 LOCATE 6+NPLACED,17:PRINT USING "\         \";TITLE$(ACTIVITY,ACTIVITY)
1250 GOTO 1280
1260 LOCATE NPLACED,28:PRINT USING "#";ACTIVITY
1270 LOCATE NPLACED,31:PRINT USING "\         \";TITLE$(ACTIVITY,ACTIVITY)
1280 CFLAG=0
1290 IF IFLAG=1 THEN GOTO 5000
1300 '
1310 IF NPLACED=N THEN GOTO 5960
1320 '
1330 'Check REL chart for heaviest relationship with placed activities
1340 'Check also for X relationships with placed activities
1350 '
1360 MAXREL=0:MINREL=-30:XFLAG=0
1370 FOR A=1 TO NPLACED
1380     FOR B=1 TO N
1390         FOR C=1 TO NPLACED
1400             IF B=TRACK(C,1) THEN GOTO 1440
1410         NEXT C
1420         IF TITLE(TRACK(A,1),B)>MAXREL THEN MAXREL=TITLE(TRACK(A,1),B):
              ACTIVITY=B:PLACED=A ELSE GOTO 1430
1430         IF TITLE(TRACK(A,1),B)=MINREL THEN XFLAG=1:TEMPACTIVITY=B:
              TEMPPLACED=A ELSE GOTO 1440
1440     NEXT B
1450 NEXT A
1460 IF XFLAG=1 THEN ACTIVITY=TEMPACTIVITY:PLACED=TEMPPLACED:GOTO 1590
      ELSE GOTO 1470
1470 IF MAXREL>17 THEN GOTO 1820
1480 '
1490 'If no A, B or X relationships with placed activities, select largest area
1500 '
1510 MAXAREA=0
1520 FOR A=1 TO N
1530    FOR B=1 TO NPLACED
1540        IF A=TRACK(B,1) THEN GOTO 1570
1550    NEXT B
1560    IF SPACE(A,1)>MAXAREA THEN ACTIVITY=A:MAXAREA=SPACE(A,1):PLACED=1
         ELSE GOTO 1570
1570 NEXT A
1580 GOTO 1820
1590 '
1600 'Place X related activity on spine in appropriate locations
1610 '
1620 IF SPACE(TRACK(PLACED,1),6)<>1 THEN GOTO 1670
1630 IF SPACE(ACTIVITY,6)=4 THEN GOTO 1650
1640 IF CORNER(3)=0 THEN XDIST=SPINE(3,1)-PLANPTS(ACTIVITY,1,2):
                         YDIST=SPINE(3,2)-PLANPTS(ACTIVITY,1,3):CORNER(3)=1:
                         SPACE(ACTIVITY,6)=3:GOTO 640 ELSE GOTO 1650
1650 IF CORNER(4)=0 THEN XDIST=SPINE(4,1)-PLANPTS(ACTIVITY,2,2):
                         YDIST=SPINE(4,2)-PLANPTS(ACTIVITY,2,3):CORNER(4)=1:
```

```
                          SPACE(ACTIVITY,6)=3:GOTO 640 ELSE GOTO 1660
1660 IF CORNER(3)=0 THEN XDIST=SPINE(3,1)-PLANPTS(ACTIVITY,1,2):
                          YDIST=SPINE(3,2)-PLANPTS(ACTIVITY,1,3):CORNER(3)=1:
                          SPACE(ACTIVITY,6)=3:GOTO 640 ELSE GOTO 4500
1670 IF SPACE(TRACK(PLACED,1),6)<>2 THEN GOTO 1720
1680 IF SPACE(ACTIVITY,6)=1 THEN GOTO 1700
1690 IF CORNER(4)=0 THEN XDIST=SPINE(4,1)-PLANPTS(ACTIVITY,2,2):
                          YDIST=SPINE(4,2)-PLANPTS(ACTIVITY,2,3):CORNER(4)=1:
                          SPACE(ACTIVITY,6)=3:GOTO 640 ELSE GOTO 1700
1700 IF CORNER(1)=0 THEN XDIST=SPINE(1,1)-PLANPTS(ACTIVITY,3,2):
                          YDIST=SPINE(1,2)-PLANPTS(ACTIVITY,3,3):CORNER(1)=1:
                          SPACE(ACTIVITY,6)=1:GOTO 640 ELSE GOTO 1710
1710 IF CORNER(4)=0 THEN XDIST=SPINE(4,1)-PLANPTS(ACTIVITY,2,2):
                          YDIST=SPINE(4,2)-PLANPTS(ACTIVITY,2,3):CORNER(4)=1:
                          SPACE(ACTIVITY,6)=3:GOTO 640 ELSE GOTO 4500
1720 IF SPACE(TRACK(PLACED,1),6)<>3 THEN GOTO 1770
1730 IF SPACE(ACTIVITY,6)=2 THEN GOTO 1750
1740 IF CORNER(1)=0 THEN XDIST=SPINE(1,1)-PLANPTS(ACTIVITY,3,2):
                          YDIST=SPINE(1,2)-PLANPTS(ACTIVITY,3,3):CORNER(1)=1:
                          SPACE(ACTIVITY,6)=1:GOTO 640 ELSE GOTO 1750
1750 IF CORNER(2)=0 THEN XDIST=SPINE(2,1)-PLANPTS(ACTIVITY,4,2):
                          YDIST=SPINE(2,2)-PLANPTS(ACTIVITY,4,3):CORNER(2)=1:
                          SPACE(ACTIVITY,6)=3:GOTO 640 ELSE GOTO 1760
1760 IF CORNER(1)=0 THEN XDIST=SPINE(1,1)-PLANPTS(ACTIVITY,3,2):
                          YDIST=SPINE(1,2)-PLANPTS(ACTIVITY,3,3):CORNER(1)=1:
                          SPACE(ACTIVITY,6)=1:GOTO 640 ELSE GOTO 4500
1770 IF SPACE(TRACK(PLACED,1),6)<>4 THEN GOTO 4500
1780 IF SPACE(ACTIVITY,6)=3 THEN GOTO 1800
1790 IF CORNER(2)=0 THEN XDIST=SPINE(2,1)-PLANPTS(ACTIVITY,4,2):
                          YDIST=SPINE(2,2)-PLANPTS(ACTIVITY,4,3):CORNER(2)=1:
                          SPACE(ACTIVITY,6)=1:GOTO 640 ELSE GOTO 1800
1800 IF CORNER(3)=0 THEN XDIST=SPINE(3,1)-PLANPTS(ACTIVITY,1,2):
                          YDIST=SPINE(3,2)-PLANPTS(ACTIVITY,1,3):CORNER(3)=1:
                          SPACE(ACTIVITY,6)=3:GOTO 640 ELSE GOTO 1810
1810 IF CORNER(2)=0 THEN XDIST=SPINE(2,1)-PLANPTS(ACTIVITY,4,2):
                          YDIST=SPINE(2,2)-PLANPTS(ACTIVITY,4,3):CORNER(2)=1:
                          SPACE(ACTIVITY,6)=1:GOTO 640 ELSE GOTO 4500
1820 '
1830 'Place activity on spine next to appropriate activity
1840 '
1850 LENGTH=ABS(PLANPTS(ACTIVITY,4,2)-PLANPTS(ACTIVITY,3,2))
1860 IF (SPACE(ACTIVITY,6)=0 OR SPACE(ACTIVITY,6)=1) AND
      SPACE(TRACK(PLACED,1),6)=1 THEN SFLAG=1:SPACE(ACTIVITY,6)=1:GOTO 2030
      ELSE GOTO 1870
1870 IF (SPACE(ACTIVITY,6)=0 OR SPACE(ACTIVITY,6)=3) AND
      SPACE(TRACK(PLACED,1),6)=3 THEN SFLAG=2:SPACE(ACTIVITY,6)=3:GOTO 2300
      ELSE GOTO 1880
1880 IF SPACE(ACTIVITY,6)=3 AND SPACE(TRACK(PLACED,1),6)=1 THEN
      SFLAG=3:GOTO 2560 ELSE GOTO 1890
1890 IF SPACE(ACTIVITY,6)=1 AND SPACE(TRACK(PLACED,1),6)=3 THEN
      SFLAG=4:GOTO 2650 ELSE GOTO 1900
1900 IF SPACE(ACTIVITY,6)=2 AND SPACE(TRACK(PLACED,1),6)=1 THEN
      SFLAG=5:GOTO 3900 ELSE GOTO 1910
1910 IF SPACE(ACTIVITY,6)=2 AND SPACE(TRACK(PLACED,1),6)=3 THEN
      SFLAG=6:GOTO 3670 ELSE GOTO 1920
1920 IF SPACE(ACTIVITY,6)=1 AND SPACE(TRACK(PLACED,1),6)=2 THEN
      SFLAG=7:GOTO 3810 ELSE GOTO 1930
1930 IF SPACE(ACTIVITY,6)=3 AND SPACE(TRACK(PLACED,1),6)=2 THEN
      SFLAG=8:GOTO 3580 ELSE GOTO 1940
1940 IF SPACE(ACTIVITY,6)=4 AND SPACE(TRACK(PLACED,1),6)=1 THEN
```

```
     SFLAG=9:GOTO 4360 ELSE GOTO 1950
1950 IF SPACE(ACTIVITY,6)=4 AND SPACE(TRACK(PLACED,1),6)=3 THEN
     SFLAG=10:GOTO 4130 ELSE GOTO 1960
1960 IF SPACE(ACTIVITY,6)=1 AND SPACE(TRACK(PLACED,1),6)=4 THEN
     SFLAG=11:GOTO 4270 ELSE GOTO 1970
1970 IF SPACE(ACTIVITY,6)=3 AND SPACE(TRACK(PLACED,1),6)=4 THEN
     SFLAG=12:GOTO 4040 ELSE GOTO 1980
1980 IF SPACE(ACTIVITY,6)=4 AND SPACE(TRACK(PLACED,1),6)=2 THEN
     SFLAG=13:GOTO 3440 ELSE GOTO 1990
1990 IF SPACE(ACTIVITY,6)=2 AND SPACE(TRACK(PLACED,1),6)=4 THEN
     SFLAG=14:GOTO 3300 ELSE GOTO 2000
2000 IF (SPACE(ACTIVITY,6)=0 OR SPACE(ACTIVITY,6)=2) AND
     SPACE(TRACK(PLACED,1),6)=2 THEN SFLAG=15:SPACE(ACTIVITY,6)=2:GOTO 2740
     ELSE GOTO 2010
2010 IF (SPACE(ACTIVITY,6)=0 OR SPACE(ACTIVITY,6)=4) AND
     SPACE(TRACK(PLACED,1),6)=4 THEN SFLAG=16:SPACE(ACTIVITY,6)=4:GOTO 3020
     ELSE GOTO 2020
2020 STOP
2030 '
2040 'East and west side check of north spine
2050 '
2060 FLAG=1:A=0:SAVEPLACED=PLACED
2070 WHILE FLAG
2080   IF TRACK(PLACED,7)<>SPINE(2,2) THEN GOTO 2250
2090   IF TRACK(PLACED,6)+LENGTH<SPINE(2,1) THEN GOTO 2170
2100   FOR B=1 TO NPLACED
2110       IF TRACK(B,7)<>SPINE(2,2) THEN GOTO 2140
2120       IF TRACK(B,8)>=TRACK(PLACED,6) AND TRACK(B,8)<=TRACK(PLACED,6)+LENGTH
            THEN GOTO 2170
2130       IF TRACK(B,8)=TRACK(PLACED,6) AND TRACK(B,6)-TRACK(B,8)>LENGTH
            THEN GOTO 2170
2140   NEXT B
2150   XDIST=TRACK(PLACED,6)-PLANPTS(ACTIVITY,4,2)
2160   YDIST=TRACK(PLACED,7)-PLANPTS(ACTIVITY,4,3):GOTO 640
2170   IF TRACK(PLACED,8)-LENGTH<SPINE(1,1) THEN GOTO 2250
2180   FOR B=1 TO NPLACED
2190       IF TRACK(B,7)<>SPINE(1,2) THEN GOTO 2220
2200       IF TRACK(B,6)<=TRACK(PLACED,8) AND TRACK(B,6)>TRACK(PLACED,8)-LENGTH
            THEN GOTO 2250
2210       IF TRACK(B,6)=TRACK(PLACED,8) AND TRACK(B,6)-TRACK(B,8)>LENGTH
            THEN GOTO 2250
2220   NEXT B
2230   XDIST=TRACK(PLACED,8)-PLANPTS(ACTIVITY,3,2)
2240   YDIST=TRACK(PLACED,9)-PLANPTS(ACTIVITY,3,3):GOTO 640
2250   IF A=0 THEN A=1:PLACED=0 ELSE GOTO 2260
2260   PLACED=PLACED+1
2270   IF PLACED>NPLACED THEN FLAG=0
2280 WEND
2290 GOTO 4500
2300 '
2310 'West and east side check of south spine
2320 '
2330  FLAG=1:A=0:SAVEPLACED=PLACED
2340  WHILE FLAG
2350    IF TRACK(PLACED,3)<>SPINE(4,2) THEN GOTO 2520
2360    IF TRACK(PLACED,2)-LENGTH<SPINE(4,1) THEN GOTO 2440
2370    FOR B=1 TO NPLACED
2380       IF TRACK(B,3)<>SPINE(4,2) THEN GOTO 2410
2390       IF TRACK(B,4)<=TRACK(PLACED,2) AND TRACK(B,4)>=TRACK(PLACED,2)-LENGTH
            THEN GOTO 2440
```

```
2400        IF TRACK(B,4)=TRACK(PLACED,2) AND TRACK(B,4)-TRACK(B,2)>LENGTH
             THEN GOTO 2440
2410     NEXT B
2420     XDIST=TRACK(PLACED,2)-PLANPTS(ACTIVITY,2,2)
2430     YDIST=TRACK(PLACED,3)-PLANPTS(ACTIVITY,2,3):GOTO 640
2440     IF TRACK(PLACED,4)+LENGTH>SPINE(3,1) THEN GOTO 2520
2450     FOR B=1 TO NPLACED
2460        IF TRACK(B,3)<>SPINE(3,2) THEN GOTO 2490
2470        IF TRACK(B,2)>=TRACK(PLACED,4) AND TRACK(B,2)<=TRACK(PLACED,4)+LENGTH
             THEN GOTO 2520
2480        IF TRACK(B,2)=TRACK(PLACED,4) AND TRACK(B,4)-TRACK(B,2)>LENGTH
             THEN GOTO 2520
2490     NEXT B
2500     XDIST=TRACK(PLACED,4)-PLANPTS(ACTIVITY,1,2)
2510     YDIST=TRACK(PLACED,5)-PLANPTS(ACTIVITY,1,3):GOTO 640
2520     IF A=0 THEN A=1:PLACED=0 ELSE GOTO 2530
2530     PLACED=PLACED+1
2540     IF PLACED>NPLACED THEN FLAG=0
2550  WEND
2560  '
2570  'South side check with north placed activity
2580  '
2590  FOR A=1 TO NPLACED
2600     IF TRACK(A,4)=160 AND TRACK(A,5)=50+WSPINE THEN GOTO 2640
2610  NEXT A
2620  XDIST=160-PLANPTS(ACTIVITY,2,2)
2630  YDIST=50+WSPINE-PLANPTS(ACTIVITY,2,3):GOTO 640
2640  GOTO 2300
2650  '
2660  'North side check with south placed activity
2670  '
2680  FOR A=1 TO NPLACED
2690     IF TRACK(A,8)=160 AND TRACK(A,9)=50 THEN GOTO 2730
2700  NEXT A
2710  XDIST=160-PLANPTS(ACTIVITY,4,2)
2720  YDIST=50-PLANPTS(ACTIVITY,4,3):GOTO 640
2730  GOTO 2030
2740  '
2750  'East side check of north and south spine
2760  '
2770  FLAG=1:A=0:SAVEPLACED=PLACED
2780  WHILE FLAG
2790     IF TRACK(PLACED,8)<160 THEN GOTO 2880
2800     IF TRACK(PLACED,6)+LENGTH>SPINE(2,1) THEN GOTO 2880
2810     FOR B=1 TO NPLACED
2820        IF TRACK(B,8)<160 THEN GOTO 2850
2830        IF TRACK(B,8)>=TRACK(PLACED,6) AND TRACK(B,8)<=TRACK(PLACED,6)+LENGTH
             THEN GOTO 2880
2840        IF TRACK(B,8)=TRACK(PLACED,6) AND TRACK(B,6)-TRACK(B,8)>LENGTH
             THEN GOTO 2880
2850     NEXT B
2860     XDIST=TRACK(PLACED,6)-PLANPTS(ACTIVITY,4,2)
2870     YDIST=TRACK(PLACED,7)-PLANPTS(ACTIVITY,4,3):GOTO 640
2880     IF TRACK(PLACED,2)<160 THEN GOTO 2970
2890     IF TRACK(PLACED,4)+LENGTH>SPINE(3,1) THEN GOTO 2970
2900     FOR B=1 TO NPLACED
2910        IF TRACK(B,2)<160 THEN GOTO 2940
2920        IF TRACK(B,2)>=TRACK(PLACED,4) AND TRACK(B,2)<=TRACK(PLACED,4)+LENGTH
             THEN GOTO 2970
2930        IF TRACK(B,2)=TRACK(PLACED,4) AND TRACK(B,4)-TRACK(B,2)>LENGTH
```

```
          THEN GOTO 2970
2940     NEXT B
2950     XDIST=TRACK(PLACED,4)-PLANPTS(ACTIVITY,1,2)
2960     YDIST=TRACK(PLACED,5)-PLANPTS(ACTIVITY,1,3):GOTO 640
2970     IF A=0 THEN A=1:PLACED=0 ELSE GOTO 2980
2980     PLACED=PLACED+1
2990     IF PLACED>NPLACED THEN FLAG=0
3000 WEND
3010 GOTO 4500
3020 '
3030 'West side check of south and north spine
3040 '
3050 FLAG=1:A=0:SAVEPLACED=PLACED
3060 WHILE FLAG
3070     IF TRACK(PLACED,6)>160 THEN GOTO 3160
3080     IF TRACK(PLACED,8)-LENGTH<SPINE(1,1) THEN GOTO 3160
3090     FOR B=1 TO NPLACED
3100        IF TRACK(B,6)>160 THEN GOTO 3130
3110        IF TRACK(B,6)<=TRACK(PLACED,8) AND TRACK(B,6)>=TRACK(PLACED,8)-LENGTH
             THEN GOTO 3160
3120        IF TRACK(B,6)=TRACK(PLACED,8) AND TRACK(B,6)-TRACK(B,8)>LENGTH
             THEN GOTO 3160
3130     NEXT B
3140     XDIST=TRACK(PLACED,8)-PLANPTS(ACTIVITY,3,2)
3150     YDIST=TRACK(PLACED,9)-PLANPTS(ACTIVITY,3,3):GOTO 640
3160     IF TRACK(PLACED,4)>160 THEN GOTO 3250
3170     IF TRACK(PLACED,2)-LENGTH<SPINE(4,1) THEN GOTO 3250
3180     FOR B=1 TO NPLACED
3190        IF TRACK(B,4)>160 THEN GOTO 3220
3200        IF TRACK(B,4)<=TRACK(PLACED,2) AND TRACK(B,4)>=TRACK(PLACED,2)-LENGTH
             THEN GOTO 3250
3210        IF TRACK(B,4)=TRACK(PLACED,2) AND TRACK(B,4)-TRACK(B,2)>LENGTH
             THEN GOTO 3250
3220     NEXT B
3230     XDIST=TRACK(PLACED,2)-PLANPTS(ACTIVITY,2,2)
3240     YDIST=TRACK(PLACED,3)=PLANPTS(ACTIVITY,2,3):GOTO 640
3250     IF A=0 THEN A=1:PLACED=0 ELSE GOTO 3260
3260     PLACED=PLACED+1
3270     IF PLACED>NPLACED THEN FLAG=0
3280 WEND
3290 GOTO 4500
3300 '
3310 'North and south side check with west placed activity
3320 '
3330 FOR A=1 TO NPLACED
3340     IF TRACK(A,8)=160 AND TRACK(A,9)=50 THEN GOTO 3380
3350 NEXT A
3360 XDIST=160-PLANPTS(ACTIVITY,4,2)
3370 YDIST=50-PLANPTS(ACTIVITY,4,3):GOTO 640
3380 FOR A=1 TO NPLACED
3390     IF TRACK(A,2)=160 AND TRACK(A,3)=50+WSPINE THEN GOTO 3430
3400 NEXT A
3410 XDIST=160-PLANPTS(ACTIVITY,1,2)
3420 YDIST=50-PLANPTS(ACTIVITY,1,3):GOTO 640
3430 GOTO 2740
3440 '
3450 'North and south side check with east placed activity
3460 '
3470 FOR A=1 TO NPLACED
3480     IF TRACK(A,6)=160 AND TRACK(A,7)=50 THEN GOTO 3520
```

```
3490 NEXT A
3500 XDIST=160-PLANPTS(ACTIVITY,3,2)
3510 YDIST=50-PLANPTS(ACTIVITY,3,3):GOTO 640
3520 FOR A=1 TO NPLACED
3530    IF TRACK(A,4)=160 AND TRACK(A,5)=50+WSPINE THEN GOTO 3570
3540 NEXT A
3550 XDIST=160-PLANPTS(ACTIVITY,2,2)
3560 YDIST=50+WSPINE-PLANPTS(ACTIVITY,2,3):GOTO 640
3570 GOTO 3020
3580 '
3590 'South side check with east placed activity
3600 '
3610 FOR A=1 TO NPLACED
3620    IF TRACK(A,4)=160 AND TRACK(A,5)=50+WSPINE THEN GOTO 3660
3630 NEXT A
3640 XDIST=160-PLANPTS(ACTIVITY,2,2)
3650 YDIST=50+WSPINE-PLANPTS(ACTIVITY,2,3):GOTO 640
3660 GOTO 2300
3670 '
3680 'East side check with south placed activity
3690 '
3700 FOR A=1 TO NPLACED
3710    IF TRACK(A,2)=160 AND TRACK(A,3)=50+WSPINE THEN GOTO 3750
3720 NEXT A
3730 XDIST=160-PLANPTS(ACTIVITY,1,2)
3740 YDIST=50+WSPINE-PLANPTS(ACTIVITY,2,2):GOTO 640
3750 FOR A=1 TO NPLACED
3760    IF TRACK(A,8)=160 AND TRACK(A,9)=50 THEN GOTO 3800
3770 NEXT A
3780 XDIST=160-PLANPTS(ACTIVITY,4,2)
3790 YDIST=50-PLANPTS(ACTIVITY,4,3):GOTO 640
3800 GOTO 2740
3810 '
3820 'North side check with east placed activity
3830 '
3840 FOR A=1 TO NPLACED
3850    IF TRACK(A,8)=160 AND TRACK(A,9)=50 THEN GOTO 3890
3860 NEXT A
3870 XDIST=160-PLANPTS(ACTIVITY,4,2)
3880 YDIST=50-PLANPTS(ACTIVITY,4,3):GOTO 640
3890 GOTO 2030
3900 '
3910 'East side check with north placed activity
3920 '
3930 FOR A=1 TO NPLACED
3940    IF TRACK(A,8)=160 AND TRACK(A,9)=50 THEN GOTO 3980
3950 NEXT A
3960 XDIST=160-PLANPTS(ACTIVITY,4,2)
3970 YDIST=50-PLANPTS(ACTIVITY,4,3):GOTO 640
3980 FOR A=1 TO NPLACED
3990    IF TRACK(A,2)=160 AND TRACK(A,3)=50+WSPINE THEN GOTO 4030
4000 NEXT A
4010 XDIST=160-PLANPTS(ACTIVITY,1,2)
4020 YDIST=50+WSPINE-PLANPTS(ACTIVITY,1,3):GOTO 640
4030 GOTO 2740
4040 '
4050 'South side check with west placed activity
4060 '
4070 FOR A=1 TO NPLACED
4080    IF TRACK(A,2)=160 AND TRACK(A,3)=50+WSPINE THEN GOTO 4120
```

```
4090 NEXT A
4100 XDIST=160-PLANPTS(ACTIVITY,1,2)
4110 YDIST=50+WSPINE-PLANPTS(ACTIVITY,1,3):GOTO 640
4120 GOTO 2300
4130 '
4140 'West side check with south placed activity
4150 '
4160 FOR A=1 TO NPLACED
4170    IF TRACK(A,6)=160 AND TRACK(A,7)=50 THEN GOTO 4210
4180 NEXT A
4190 XDIST=160-PLANPTS(ACTIVITY,3,2)
4200 YDIST=50-PLANPTS(ACTIVITY,3,3):GOTO 640
4210 FOR A=1 TO NPLACED
4220    IF TRACK(A,4)=160 AND TRACK(A,5)=50+WSPINE THEN GOTO 4260
4230 NEXT A
4240 XDIST=160-PLANPTS(ACTIVITY,2,2)
4250 YDIST=50+WSPINE-PLANPTS(ACTIVITY,2,3):GOTO 640
4260 GOTO 3020
4270 '
4280 'North side check with west placed activity
4290 '
4300 FOR A=1 TO NPLACED
4310    IF TRACK(A,8)=160 AND TRACK(A,9)=50 THEN GOTO 4350
4320 NEXT A
4330 XDIST=160-PLANPTS(ACTIVITY,3,2)
4340 YDIST=50-PLANPTS(ACTIVITY,3,3):GOTO 640
4350 GOTO 2030
4360 '
4370 'West side check with north placed activity
4380 '
4390 FOR A=1 TO NPLACED
4400    IF TRACK(A,6)=160 AND TRACK(A,7)=50 THEN GOTO 4440
4410 NEXT A
4420 XDIST=160-PLANPTS(ACTIVITY,3,2)
4430 YDIST=50-PLANPTS(ACTIVITY,3,3):GOTO 640
4440 FOR A=1 TO NPLACED
4450    IF TRACK(A,4)=160 AND TRACK(A,5)=50+WSPINE THEN GOTO 4490
4460 NEXT A
4470 XDIST=160-PLANPTS(ACTIVITY,2,2)
4480 YDIST=50-PLANPTS(ACTIVITY,2,3):GOTO 640
4490 GOTO 3020
4500 '
4510 'No locations available on desired spine side
4520 '
4530 XDIST=250-PLANPTS(ACTIVITY,1,2)
4540 YDIST=150-PLANPTS(ACTIVITY,1,3)
4550 FOR A=1 TO SPACE(ACTIVITY,11)
4560    PLANPTS(ACTIVITY,A,2)=PLANPTS(ACTIVITY,A,2)+XDIST
4570    PLANPTS(ACTIVITY,A,3)=PLANPTS(ACTIVITY,A,3)+YDIST
4580 NEXT A
4590 FLAG=1:I=1:COUNT=0:PLACED=SAVEPLACED
4600 WHILE FLAG
4610    I=PLANPTS(ACTIVITY,I,1)
4620    LINE (PLANPTS(ACTIVITY,PLANPTS(ACTIVITY,I,1),2),PLANPTS(ACTIVITY,
          PLANPTS(ACTIVITY,I,1),3))-(PLANPTS(ACTIVITY,PLANPTS(ACTIVITY,I,4),2)
          ,PLANPTS(ACTIVITY,PLANPTS(ACTIVITY,I,4),3)),1
4630    I=PLANPTS(ACTIVITY,I,4)
4640    COUNT=COUNT+1
4650    IF COUNT>=SPACE(ACTIVITY,11) THEN FLAG=0
4660 WEND
```

```
4670 LOCATE 21,1:PRINT USING "#";ACTIVITY:LOCATE 21,4:
      PRINT TITLE$(ACTIVITY,ACTIVITY)
4680 LOCATE 22,1:PRINT TITLE$(ACTIVITY,TRACK(PLACED,1))
4690 LOCATE 22,2:PRINT "-RELATIONSHIP WITH ";TITLE$(TRACK(PLACED,1),
      TRACK(PLACED,1))
4700 LOCATE 23,1:PRINT "LOCATE ON SPINE"
4710 FOR A=1 TO 500:NEXT A
4720 PEN ON
4730 IF PEN(3)=0 THEN GOTO 4720
4740 X1=PEN(4):Y1=PEN(5)
4750 PSET (X1,Y1)
4760 LOCATE 21,1:PRINT SPC(30)
4770 LOCATE 22,1:PRINT SPC(30)
4780 LOCATE 23,1:PRINT "CORRECT POINT?      YES    NO      "
4790 LINE (150,185)-(175,170),1,B:LINE (195,185)-(220,170),1,B
4800 PEN ON
4810 IF PEN(3)=0 THEN GOTO 4800
4820 IF PEN(8)>21 AND PEN(9)>24 THEN PSET (X1,Y1),0:GOTO 4710 ELSE GOTO 4830
4830 PSET (X1,Y1),0
4840 IF Y1<=50 THEN XDIST=X1-PLANPTS(ACTIVITY,4,2):
                    YDIST=50-PLANPTS(ACTIVITY,4,3):SPACE(ACTIVITY,6)=1
                    ELSE GOTO 4850
4850 IF Y1>=50+WSPINE THEN XDIST=X1-PLANPTS(ACTIVITY,1,2):
                    YDIST=50+WSPINE-PLANPTS(ACTIVITY,1,3):SPACE(ACTIVITY,6)=3
                    ELSE GOTO 4860
4860 '
4870 'Erase activity
4880 '
4890 FLAG=1:I=1:COUNT=0
4900 WHILE FLAG
4910     I=PLANPTS(ACTIVITY,I,1)
4920     LINE (PLANPTS(ACTIVITY,PLANPTS(ACTIVITY,I,1),2),PLANPTS(ACTIVITY,
           PLANPTS(ACTIVITY,I,1),3))-(PLANPTS(ACTIVITY,PLANPTS(ACTIVITY,I,4),2)
           ,PLANPTS(ACTIVITY,PLANPTS(ACTIVITY,I,4),3)),0
4930     I=PLANPTS(ACTIVITY,I,4)
4940     COUNT=COUNT+1
4950     IF COUNT>=SPACE(ACTIVITY,11) THEN FLAG=0
4960 WEND
4970 LOCATE 23,1:PRINT SPC(39)
4980 LINE (150,185)-(170,195),0,B:LINE (195,185)-(220,170),0,B
4990 IFLAG=1:GOTO 640
5000 '
5010 'Modify location of activity
5020 '
5030 CFLAG=0
5040 LOCATE 23,1:PRINT "ANY CHANGES?        YES    NO      "
5050 LINE (150,185)-(175,170),1,B:LINE (195,185)-(220,170),1,B
5060 PEN ON
5070 IF PEN(3)=0 THEN GOTO 5060
5080 IF PEN(8)>21 AND PEN(9)>24 THEN GOTO 1300
5090 LINE (150,185)-(175,170),0,B:LINE (195,185)-(220,170),0,B
5100 LOCATE 23,1:PRINT "  MODIFY    REDRAW                 "
5110 LINE (5,185)-(75,170),1,B:LINE (80,185)-(150,170),1,B
5120 PEN ON
5130 IF PEN(3)=0 THEN GOTO 5120
5140 IF PEN(9)>10 THEN GOTO 5730
5150 LINE (5,185)-(75,170),0,B:LINE (80,185)-(150,170),0,B
5160 LOCATE 23,1:PRINT SPC(39)
5170 LOCATE 21,1:PRINT "INDICATE NEW ACTIVITY LOCATIONS  "
5180 FOR A=1 TO 500:NEXT A
```

```
5190 PEN ON
5200 IF PEN(3)=0 THEN GOTO 5190
5210 X1=PEN(4):Y1=PEN(5)
5220 PSET (X1,Y1)
5230 LOCATE 23,1:PRINT "CORRECT POINT?      YES    NO      "
5240 LINE (150,185)-(175,170),1,B:LINE (195,185)-(220,170),1,B
5250 PEN ON
5260 IF PEN(3)=0 THEN GOTO 5250
5270 IF PEN(8)>21 AND PEN(9)>24 THEN PSET (X1,Y1),0:GOTO 5180 ELSE GOTO 5280
5280 PSET (X1,Y1),0
5290 '
5300 'Check TRACK array for appropriate activity coodinates
5310 '
5320 FOR A=1 TO N
5330    IF X1>=PLANPTS(A,1,2) AND X1<=PLANPTS(A,2,2) AND Y1>=PLANPTS(A,1,3) AND
         Y1<=PLANPTS(A,4,3) THEN ACTIVITY=A:GOTO 5360 ELSE GOTO 5340
5340 NEXT A
5350 STOP
5360 FOR A=1 TO NPLACED
5370    IF ACTIVITY=TRACK(A,1) THEN GOTO 5400
5380 NEXT A
5390 STOP
5400 LOCATE 1,38:PRINT "*"
5410 FOR B=1 TO 500:NEXT B
5420 PEN ON
5430 IF PEN(3)=0 THEN GOTO 5420
5440 X2=PEN(4):Y2=PEN(5)
5450 PSET (X2,Y2)
5460 FOR B=1 TO 500:NEXT B
5470 PEN ON
5480 IF PEN(3)=0 THEN GOTO 5470
5490 IF PEN(8)>21 AND PEN(9)>24 THEN PSET (X2,Y2),0:GOTO 5410 ELSE GOTO 5500
5500 PSET (X2,Y2),0
5510 LOCATE 1,38:PRINT "  "
5520 FOR B=1 TO 500:NEXT B
5530 XDIST=X2-PLANPTS(ACTIVITY,1,2)
5540 IF PLANPTS(ACTIVITY,4,3)=50 AND Y2<=50+WSPINE/2 THEN YDIST=0
5550 IF PLANPTS(ACTIVITY,4,3)=50 AND Y2> 50+WSPINE/2 THEN
      YDIST=PLANPTS(ACTIVITY,3,3)-PLANPTS(ACTIVITY,2,3)+WSPINE:
      SPACE(ACTIVITY,6)=3 ELSE GOTO 5560
5560 IF PLANPTS(ACTIVITY,1,3)=50+WSPINE AND Y2<=50+WSPINE/2 THEN
      YDIST=PLANPTS(ACTIVITY,2,3)-PLANPTS(ACTIVITY,3,3)-WSPINE:
      SPACE(ACTIVITY,6)=1 ELSE GOTO 5570
5570 IF PLANPTS(ACTIVITY,1,3)=50+WSPINE AND Y2> 50+WSPINE/2 THEN YDIST=0
5580 '
5590 'Erase existing activity
5600 '
5610 XPLACE=INT(PLANPTS(ACTIVITY,1,2)/8):YPLACE=INT(PLANPTS(ACTIVITY,1,3)/8)
5620 IF TRACK(A,3)=50 THEN LOCATE YPLACE,XPLACE+1:PRINT "  ":GOTO 5640
      ELSE GOTO 5630
5630 IF TRACK(A,3)=50+WSPINE THEN LOCATE YPLACE+5,XPLACE+1:PRINT "  ":GOTO 5640
      ELSE STOP
5640 FLAG=1:COUNT=0:I=1
5650 WHILE FLAG
5660    I=PLANPTS(ACTIVITY,I,1)
5670       LINE (PLANPTS(ACTIVITY,PLANPTS(ACTIVITY,I,1),2),PLANPTS(ACTIVITY,
           PLANPTS(ACTIVITY,I,1),3))-(PLANPTS(ACTIVITY,PLANPTS(ACTIVITY,I,4),2),
           PLANPTS(ACTIVITY,PLANPTS(ACTIVITY,I,4),3)),0
5680    I=PLANPTS(ACTIVITY,I,4)
5690    COUNT=COUNT+1
```

```
5700    IF COUNT>=SPACE(ACTIVITY,11) THEN FLAG=0
5710 WEND
5720 CFLAG=1:GOTO 640
5730 '
5740 'Redraw screen
5750 '
5760 LINE (5,185)-(75,170),0,B:LINE (80,185)-(150,170),0,B
5770 LOCATE 23,1:PRINT SPC(39)
5780 FOR A=1 TO NPLACED
5790   FLAG=1:COUNT=0:I=1
5800   WHILE FLAG
5810     I=PLANPTS(TRACK(A,1),I,1)
5820     LINE (PLANPTS(TRACK(A,1),PLANPTS(TRACK(A,1),I,1),2),PLANPTS(TRACK(A,1),
          PLANPTS(TRACK(A,1),I,1),3))-(PLANPTS(TRACK(A,1),PLANPTS(TRACK(A,1),I,4
          ),2),PLANPTS(TRACK(A,1),PLANPTS(TRACK(A,1),I,4),3)),1
5830     I=PLANPTS(TRACK(A,1),I,4)
5840     COUNT=COUNT+1
5850     IF COUNT>=SPACE(TRACK(A,1),11) THEN FLAG=0
5860   WEND
5870 NEXT A
5880 LINE ((320-LSPINE)/2,50)-((320-LSPINE)/2+LSPINE,50+WSPINE),1,B
5890 GOTO 5000
5900 '
5910 'Allow another circulation pattern design to be developed
5920 '
5930 LOCATE 21,1:PRINT SPC(39)
5940 LINE (150,185)-(175,170),0,B:LINE (195,185)-(220,170),0,B
5950 LOCATE 23,1:PRINT SPC(39)
5960 LOCATE 23,1:PRINT "SAVE PLAN?          YES    NO      "
5970 LINE (150,185)-(175,170),1,B:LINE (195,185)-(220,170),1,B
5980 PEN ON
5990 IF PEN(3)=0 THEN GOTO 5980
6000 IF PEN(8)>21 AND PEN(9)<23 THEN JUMP$="LINE":GOTO 6260 ELSE GOTO 6010
6010 LOCATE 23,1:PRINT "PLACE ON SITE?      YES    NO      "
6020 LINE (150,185)-(175,170),1,B:LINE (195,185)-(220,170),1,B
6030 PEN ON
6040 IF PEN(3)=0 THEN GOTO 6030
6050 IF PEN(8)>21 AND PEN(9)<23 THEN GOTO 6250
6060 LOCATE 23,1:PRINT "ANOTHER DESIGN?     YES    NO      "
6070 LINE (150,185)-(175,170),1,B:LINE (195,185)-(220,170),1,B
6080 PEN ON
6090 IF PEN(3)=0 THEN GOTO 6080
6100 IF PEN(8)>21 AND PEN(9)>24 THEN GOTO 6170
6110 LOCATE 23,1:PRINT "UPDATE DATA?        YES    NO      "
6120 LINE (150,185)-(175,170),1,B:LINE (195,185)-(220,170),1,B
6130 PEN ON
6140 IF PEN(3)=0 THEN GOTO 6130
6150 IF PEN(8)>21 AND PEN(9)>24 THEN GOTO 6240
6160 GOTO 6230
6170 LOCATE 23,1:PRINT "START AGAIN?        YES    NO      "
6180 LINE (150,185)-(175,170),1,B:LINE (195,185)-(220,170),1,B
6190 PEN ON
6200 IF PEN(3)=0 THEN GOTO 6190
6210 IF PEN(8)>21 AND PEN(9)<23 THEN GOTO 6220 ELSE STOP
6220 CHAIN MERGE "CREATE",10,DELETE 10-6210
6230 CHAIN MERGE "SPACE",10,ALL,DELETE 10-6220
6240 CHAIN MERGE "GEOMTRY",10,ALL,DELETE 10-6230
6250 CHAIN MERGE "SITE",10,ALL,DELETE 10-6240
6260 CHAIN MERGE "PLNFILE",10,ALL,DELETE 10-6250
```

```
10 '*****************************************************************
20 'COURTYARD Subroutine
30 '*****************************************************************
40 'This routine allows the user to input a physical description of the
50 'activities.  Inputs include area, proportion, ceiling height, and
60 'wall assignment.  User may modify shape using drafting routine.
70 '
80 CLS:PEN OFF
90 FOR A=1 TO 4
100       CORNER(A)=0
110 NEXT A
120 CFLAG=0:JUMP=2
130 LOCATE 2,13:PRINT "POINT Subroutine"
140 LOCATE 8,1  :PRINT "   SQUARE"
150 LOCATE 14,1 :PRINT "   OFFSET"
160 LINE (10,50)-(90,65),1,B:LINE (10,100)-(90,115),1,B
170 LINE (120,50)-(135,65),1,B
180 LINE(120,100)-(135,115),1,B:LINE(135,115)-(150,130),1,B
190 PEN ON
200 IF PEN(3)=0 THEN GOTO 190
210 IF PEN(8)>=7 AND PEN(8)<=11 AND PEN(9)<=15 THEN GOTO 240
220 IF PEN(8)>11 AND PEN(8)<=15 AND PEN(9)<=15 THEN GOTO 230
230 STOP
240 '
250 'Square courtyard placement
260 '
270 LOCATE 23,1:INPUT "ENTER COURTYARD DIMENSION  ",YARD
280 YARD=YARD*SCALE
290 CLS
300 LOCATE 23,1:PRINT "INTERACTIVE ?      YES   NO"
310 LINE (150,185)-(175,170),1,B:LINE (195,185)-(220,170),1,B
320 PEN ON
330 IF PEN(3)=0 THEN GOTO 320
340 IF PEN(8)>21 AND PEN(9)>24 THEN IFLAG=0 ELSE IFLAG=1
350 CLS
360 '
370 'Draw skeleton courtyard
380 '
390 YARDPTS(1,1)=160-YARD/2:YARDPTS(1,2)=50
400 YARDPTS(2,1)=160+YARD/2:YARDPTS(2,2)=50
410 YARDPTS(3,1)=160+YARD/2:YARDPTS(3,2)=50+YARD
420 YARDPTS(4,1)=160-YARD/2:YARDPTS(4,2)=50+YARD
430 LINE (YARDPTS(1,1),YARDPTS(1,2))-(YARDPTS(3,1).YARDPTS(3,2)),1,B
440 '
450 'Determine activity with largest square footage
460 '
470 MAXAREA=0:NPLACED=0
480 FOR A=1 TO N
490    IF SPACE(A,1)>MAXAREA THEN ACTIVITY=A:MAXAREA=SPACE(A,1) ELSE GOTO 500
500 NEXT A
510 SIDE=ABS(PLANPTS(ACTIVITY,2,3)-PLANPTS(ACTIVITY,3,3))
520 LENGTH=ABS(PLANPTS(ACTIVITY,1,2)-PLANPTS(ACTIVITY,2,2))
530 '
540 'Check wall assignment and determine initial activity placement
550 '
560 IF SPACE(ACTIVITY,6)=4 THEN GOTO 660
570 IF SPACE(ACTIVITY,6)=3 THEN GOTO 640
580 IF SPACE(ACTIVITY,6)=2 THEN GOTO 620
590 IF SPACE(ACTIVITY,6)=0 THEN SPACE(ACTIVITY,6)=1
600 XDIST=YARDPTS(1,1)-PLANPTS(ACTIVITY,4,2):
```

```
    YDIST=YARDPTS(1,2)-PLANPTS(ACTIVITY,4,3)
610 GOTO 670
620 XDIST=YARDPTS(2,1)-PLANPTS(ACTIVITY,1,2):
    YDIST=YARDPTS(2,2)-PLANPTS(ACTIVITY,1,3)
630 GOTO 670
640 XDIST=YARDPTS(3,1)-PLANPTS(ACTIVITY,2,2):
    YDIST=YARDPTS(3,2)-PLANPTS(ACTIVITY,2,3)
650 GOTO 670
660 XDIST=YARDPTS(4,1)-PLANPTS(ACTIVITY,3,2):
    YDIST=YARDPTS(4,2)-PLANPTS(ACTIVITY,3,3)
670 '
680 'Translate activity coordinates to courtyard
690 '
700 FOR A=1 TO SPACE(ACTIVITY,11)
710      PLANPTS(ACTIVITY,A,2)=PLANPTS(ACTIVITY,A,2)+XDIST
720      PLANPTS(ACTIVITY,A,3)=PLANPTS(ACTIVITY,A,3)+YDIST
730 NEXT A
740 IF PLANPTS(ACTIVITY,3,2)>YARDPTS(2,1) AND
       PLANPTS(ACTIVITY,3,3)=YARDPTS(2,2) THEN CORNER(2)=1
750 IF PLANPTS(ACTIVITY,4,2)=YARDPTS(3,1) AND
       PLANPTS(ACTIVITY,4,3)>YARDPTS(3,2) THEN CORNER(3)=1
760 IF PLANPTS(ACTIVITY,1,2)<YARDPTS(4,1) AND
       PLANPTS(ACTIVITY,1,3)=YARDPTS(4,2) THEN CORNER(4)=1
770 IF PLANPTS(ACTIVITY,2,2)=YARDPTS(1,1) AND
       PLANPTS(ACTIVITY,2,3)<YARDPTS(1,2) THEN CORNER(1)=1
780 '
790 'Draw activity on courtyard
800 '
810   FLAG=1:I=1:COUNT=0
820   IF CFLAG=0 THEN NPLACED=NPLACED+1
830    WHILE FLAG
840       I=PLANPTS(ACTIVITY,I,1)
850       LINE (PLANPTS(ACTIVITY,PLANPTS(ACTIVITY,I,1),2),PLANPTS(ACTIVITY,
           PLANPTS(ACTIVITY,I,1),3))-(PLANPTS(ACTIVITY,PLANPTS(ACTIVITY,I,4),2)
           ,PLANPTS(ACTIVITY,PLANPTS(ACTIVITY,I,4),3)),1
860       I=PLANPTS(ACTIVITY,I,4)
870       COUNT=COUNT+1
880       IF COUNT>=SPACE(ACTIVITY,11) THEN FLAG=0
890    WEND
900 '
910 'Store reference points located on courtyard
920 '
930 IF CFLAG=0 THEN GOTO 980
940 FOR A=1 TO NPLACED
950     IF ACTIVITY=TRACK(A,1) THEN M=A:GOTO 1000 ELSE GOTO 960
960 NEXT A
970 STOP
980 M=NPLACED
990 TRACK(M,1)=ACTIVITY
1000 IF PLANPTS(ACTIVITY,4,3)=50 AND PLANPTS(ACTIVITY,3,3)=50
      THEN GOTO 1010 ELSE GOTO 1050
1010 TRACK(M,2)=0                        :TRACK(M,3)=0
1020 TRACK(M,4)=0                        :TRACK(M,5)=0
1030 TRACK(M,6)=PLANPTS(ACTIVITY,3,2):TRACK(M,7)=PLANPTS(ACTIVITY,3,3)
1040 TRACK(M,8)=PLANPTS(ACTIVITY,4,2):TRACK(M,9)=PLANPTS(ACTIVITY,4,3):GOTO 1200

1050 IF PLANPTS(ACTIVITY,1,3)=50+YARD AND PLANPTS(ACTIVITY,2,3)=50+YARD
      THEN GOTO 1060 ELSE GOTO 1100
1060 TRACK(M,2)=PLANPTS(ACTIVITY,1,2):TRACK(M,3)=PLANPTS(ACTIVITY,1,3)
1070 TRACK(M,4)=PLANPTS(ACTIVITY,2,2):TRACK(M,5)=PLANPTS(ACTIVITY,2,3)
```

```
1080 TRACK(M,6)=0                        :TRACK(M,7)=0
1090 TRACK(M,8)=0                        :TRACK(M,9)=0:GOTO 1200
1100 IF PLANPTS(ACTIVITY,1,2)=160+YARD/2 AND PLANPTS(ACTIVITY,4,2)=160+YARD/2
      THEN GOTO 1110 ELSE GOTO 1150
1110 TRACK(M,2)=PLANPTS(ACTIVITY,1,2):TRACK(M,3)=PLANPTS(ACTIVITY,1,3)
1120 TRACK(M,4)=0                        :TRACK(M,5)=0
1130 TRACK(M,6)=0                        :TRACK(M,7)=0
1140 TRACK(M,8)=PLANPTS(ACTIVITY,4,2):TRACK(M,9)=PLANPTS(ACTIVITY,4,3):GOTO 1200

1150 IF PLANPTS(ACTIVITY,2,2)=160-YARD/2 AND PLANPTS(ACTIVITY,3,2)=160-YARD/2
      THEN GOTO 1160 ELSE STOP
1160 TRACK(M,2)=0                        :TRACK(M,3)=0
1170 TRACK(M,4)=PLANPTS(ACTIVITY,2,2):TRACK(M,5)=PLANPTS(ACTIVITY,2,3)
1180 TRACK(M,6)=PLANPTS(ACTIVITY,3,2):TRACK(M,7)=PLANPTS(ACTIVITY,3,3)
1190 TRACK(M,8)=0                        :TRACK(M,9)=0:GOTO 1200
1200 '
1210 'Label floor plan
1220 '
1230 XPLACE=INT(PLANPTS(ACTIVITY,1,2)/8):YPLACE=INT(PLANPTS(ACTIVITY,1,3)/8)
1240 IF TRACK(M,7)=YARDPTS(1,2) AND TRACK(M,9)=YARDPTS(1,2) THEN
      LOCATE YPLACE,XPLACE+1:GOTO 1280 ELSE GOTO 1250
1250 IF TRACK(M,2)=YARDPTS(2,1) AND TRACK(M,8)=YARDPTS(2,1) THEN
      LOCATE YPLACE,XPLACE+5:GOTO 1280 ELSE GOTO 1260
1260 IF TRACK(M,3)=YARDPTS(3,2) AND TRACK(M,5)=YARDPTS(3,2) THEN
      LOCATE YPLACE+5,XPLACE+1:GOTO 1280 ELSE GOTO 1270
1270 IF TRACK(M,4)=YARDPTS(4,1) AND TRACK(M,6)=YARDPTS(4,1) THEN
      LOCATE YPLACE,XPLACE+1:GOTO 1280 ELSE STOP
1280 IF ACTIVITY< 10 THEN PRINT USING "#"; ACTIVITY
1290 IF ACTIVITY>=10 THEN PRINT USING "##";ACTIVITY
1300 IF CFLAG=1 THEN GOTO 1420
1310 IF NPLACED<= 6 THEN GOTO 1340
1320 IF NPLACED<=12 THEN GOTO 1370
1330 IF NPLACED<=20 THEN GOTO 1400
1340 LOCATE 12+NPLACED,1:PRINT USING "#";ACTIVITY
1350 LOCATE 12+NPLACED,3:PRINT USING "\         \";TITLE$(ACTIVITY,ACTIVITY)
1360 GOTO 1420
1370 LOCATE 6+NPLACED,14:PRINT USING "#";ACTIVITY
1380 LOCATE 6+NPLACED,17:PRINT USING "\         \";TITLE$(ACTIVITY,ACTIVITY)
1390 GOTO 1420
1400 LOCATE NPLACED,28:PRINT USING "#";ACTIVITY
1410 LOCATE NPLACED,31:PRINT USING "\         \";TITLE$(ACTIVITY,ACTIVITY)
1420 CFLAG=0
1430 IF IFLAG=1 THEN GOTO 4380
1440 '
1450 IF NPLACED=N THEN GOTO 5420
1460 '
1470 'Check REL chart for heaviest relationship with placed activities
1480 'Check also for X relationships with placed activities
1490 '
1500 MAXREL=0:MINREL=-30:XFLAG=0
1510 FOR A=1 TO NPLACED
1520     FOR B=1 TO N
1530         FOR C=1 TO NPLACED
1540             IF B=TRACK(C,1) THEN GOTO 1580
1550         NEXT C
1560         IF TITLE(TRACK(A,1),B)>MAXREL THEN MAXREL=TITLE(TRACK(A,1),B):
              ACTIVITY=B:PLACED=A ELSE GOTO 1570
1570         IF TITLE(TRACK(A,1),B)=MINREL THEN XFLAG=1:TEMPACTIVITY=B:
              TEMPPLACED=A ELSE GOTO 1580
1580     NEXT B
```

```
1590 NEXT A
1600 IF XFLAG=1 THEN ACTIVITY=TEMPACTIVITY:PLACED=TEMPPLACED:GOTO 1730
     ELSE GOTO 1610
1610 IF MAXREL>17 THEN GOTO 1970
1620 '
1630 'If no A, B or X relationships with placed activities, select largest area
1640 '
1650 MAXAREA=0
1660 FOR A=1 TO N
1670    FOR B=1 TO NPLACED
1680        IF A=TRACK(B,1) THEN GOTO 1710
1690    NEXT B
1700    IF SPACE(A,1)>MAXAREA THEN ACTIVITY=A:MAXAREA=SPACE(A,1):PLACED=1
        ELSE GOTO 1710
1710 NEXT A
1720 GOTO 1970
1730 '
1740 'Place X related activity on spine in appropriate locations
1750 '
1760 IF SPACE(TRACK(PLACED,1),6)<>1 THEN GOTO 1810
1770 IF SPACE(ACTIVITY,6)=4 THEN GOTO 1790
1780 IF CORNER(3)=0 THEN XDIST=YARDPTS(3,1)-PLANPTS(ACTIVITY,1,2):
                         YDIST=YARDPTS(3,2)-PLANPTS(ACTIVITY,1,3):CORNER(3)=1:
                         SPACE(ACTIVITY,6)=3:GOTO 670 ELSE GOTO 1790
1790 IF CORNER(4)=0 THEN XDIST=YARDPTS(4,1)-PLANPTS(ACTIVITY,2,2):
                         YDIST=YARDPTS(4,2)-PLANPTS(ACTIVITY,2,3):CORNER(4)=1:
                         SPACE(ACTIVITY,6)=3:GOTO 670 ELSE GOTO 1800
1800 IF CORNER(3)=0 THEN XDIST=YARDPTS(3,1)-PLANPTS(ACTIVITY,1,2):
                         YDIST=YARDPTS(3,2)-PLANPTS(ACTIVITY,1,3):CORNER(3)=1:
                         SPACE(ACTIVITY,6)=3:GOTO 670 ELSE GOTO 1810
1810 IF SPACE(TRACK(PLACED,1),6)<>2 THEN GOTO 1860
1820 IF SPACE(ACTIVITY,6)=1 THEN GOTO 1840
1830 IF CORNER(4)=0 THEN XDIST=YARDPTS(4,1)-PLANPTS(ACTIVITY,2,2):
                         YDIST=YARDPTS(4,2)-PLANPTS(ACTIVITY,2,3):CORNER(4)=1:
                         SPACE(ACTIVITY,6)=3:GOTO 670 ELSE GOTO 1840
1840 IF CORNER(1)=0 THEN XDIST=YARDPTS(1,1)-PLANPTS(ACTIVITY,3,2):
                         YDIST=YARDPTS(1,2)-PLANPTS(ACTIVITY,3,3):CORNER(1)=1:
                         SPACE(ACTIVITY,6)=1:GOTO 670 ELSE GOTO 1850
1850 IF CORNER(4)=0 THEN XDIST=YARDPTS(4,1)-PLANPTS(ACTIVITY,2,2):
                         YDIST=YARDPTS(4,2)_PLANPTS(ACTIVITY,2,3):CORNER(4)=1:
                         SPACE(ACTIVITY,6)=3:GOTO 670 ELSE GOTO 1860
1860 IF SPACE(TRACK(PLACED,1),6)<>3 THEN GOTO 1910
1870 IF SPACE(ACTIVITY,6)=2 THEN GOTO 1890
1880 IF CORNER(1)=0 THEN XDIST=YARDPTS(1,1)-PLANPTS(ACTIVITY,3,2):
                         YDIST=YARDPTS(1,2)-PLANPTS(ACTIVITY,3,3):CORNER(1)=1:
                         SPACE(ACTIVITY,6)=1:GOTO 670 ELSE GOTO 1890
1890 IF CORNER(2)=0 THEN XDIST=YARDPTS(2,1)-PLANPTS(ACTIVITY,4,2):
                         YDIST=YARDPTS(2,2)-PLANPTS(ACTIVITY,4,3):CORNER(2)=1:
                         SPACE(ACTIVITY,6)=1:GOTO 670 ELSE GOTO 1900
1900 IF CORNER(1)=0 THEN XDIST=YARDPTS(1,1)-PLANPTS(ACTIVITY,3,2):
                         YDIST=YARDPTS(1,2)-PLANPTS(ACTIVITY,3,3):CORNER(1)=1:
                         SPACE(ACTIVITY,6)=1:GOTO 670 ELSE GOTO 1910
1910 IF SPACE(TRACK(PLACED,1),6)<>4 THEN GOTO 1960
1920 IF SPACE(ACTIVITY,6)=3 THEN GOTO 1940
1930 IF CORNER(2)=0 THEN XDIST=YARDPTS(2,1)-PLANPTS(ACTIVITY,4,2):
                         YDIST=YARDPTS(2,2)-PLANPTS(ACTIVITY,4,3):CORNER(2)=1:
                         SPACE(ACTIVITY,6)=1:GOTO 670 ELSE GOTO 1940
1940 IF CORNER(3)=0 THEN XDIST=YARDPTS(3,1)-PLANPTS(ACTIVITY,1,2):
                         YDIST=YARDPTS(3,2)-PLANPTS(ACTIVITY,1,3):CORNER(3)=1:
                         SPACE(ACTIVITY,6)=3:GOTO 670 ELSE GOTO 1950
```

```
1950 IF CORNER(2)=0 THEN XDIST=YARDPTS(2,1)-PLANPTS(ACTIVITY,4,2):
                         YDIST=YARDPTS(2,2)-PLANPTS(ACTIVITY,4,3):CORNER(2)=1:
                         SPACE(ACTIVITY,6)=1:GOTO 670 ELSE GOTO 1960
1960 GOTO 3890
1970 '
1980 'Place activity on courtyard next to appropriate activity
1990 '
2000 NLENGTH=ABS(PLANPTS(ACTIVITY,4,2)-PLANPTS(ACTIVITY,3,2))
2010 ELENGTH=ABS(PLANPTS(ACTIVITY,1,3)-PLANPTS(ACTIVITY,4,3))
2020 SLENGTH=ABS(PLANPTS(ACTIVITY,1,2)-PLANPTS(ACTIVITY,2,2))
2030 WLENGTH=ABS(PLANPTS(ACTIVITY,2,3)-PLANPTS(ACTIVITY,3,3))
2040 '
2050 IF (SPACE(ACTIVITY,6)=0 OR SPACE(ACTIVITY,6)=1) AND
      SPACE(TRACK(PLACED,1),6)=1 THEN SFLAG=1:SPACE(ACTIVITY,6)=1:GOTO 2240
      ELSE GOTO 2060
2060 IF (SPACE(ACTIVITY,6)=0 OR SPACE(ACTIVITY,6)=3) AND
      SPACE(TRACK(PLACED,1),6)=3 THEN SFLAG=2:SPACE(ACTIVITY,6)=3:GOTO 2480
      ELSE GOTO 2070
2070 IF SPACE(ACTIVITY,6)=3 AND SPACE(TRACK(PLACED,1),6)=1 THEN
      SFLAG=3:GOTO 2720 ELSE GOTO 2080
2080 IF SPACE(ACTIVITY,6)=1 AND SPACE(TRACK(PLACED,1),6)=3 THEN
      SFLAG=4:GOTO 2890 ELSE GOTO 2090
2090 IF (SPACE(ACTIVITY,6)=0 OR SPACE(ACTIVITY,6)=2) AND
      SPACE(TRACK(PLACED,1),6)=2 THEN SFLAG=5:SPACE(ACTIVITY,6)=2:GOTO 3060
      ELSE GOTO 2100
2100 IF (SPACE(ACTIVITY,6)=0 OR SPACE(ACTIVITY,6)=4) AND
      SPACE(TRACK(PLACED,1),6)=4 THEN SFLAG=6:SPACE(ACTIVITY,6)=4:GOTO 3300
      ELSE GOTO 2110
2110 IF SPACE(ACTIVITY,6)=4 AND SPACE(TRACK(PLACED,1),6)=2 THEN
      SFLAG=7:GOTO 3540 ELSE GOTO 2120
2120 IF SPACE(ACTIVITY,6)=2 AND SPACE(TRACK(PLACED,1),6)=4 THEN
      SFLAG=8:GOTO 3710 ELSE GOTO 2130
2130 IF SPACE(ACTIVITY,6)=1 AND SPACE(TRACK(PLACED,1),6)=4 THEN
      SFLAG=9:GOTO 2970 ELSE GOTO 2140
2140 IF SPACE(ACTIVITY,6)=1 AND SPACE(TRACK(PLACED,1),6)=2 THEN
      SFLAG=10:GOTO 2970 ELSE GOTO 2150
2150 IF SPACE(ACTIVITY,6)=3 AND SPACE(TRACK(PLACED,1),6)=4 THEN
      SFLAG=11:GOTO 2720 ELSE GOTO 2160
2160 IF SPACE(ACTIVITY,6)=3 AND SPACE(TRACK(PLACED,1),6)=2 THEN
      SFLAG=12:GOTO 2720 ELSE GOTO 2170
2170 IF SPACE(ACTIVITY,6)=4 AND SPACE(TRACK(PLACED,1),6)=3 THEN
      SFLAG=13:GOTO 3540 ELSE GOTO 2180
2180 IF SPACE(ACTIVITY,6)=4 AND SPACE(TRACK(PLACED,1),6)=1 THEN
      SFLAG=14:GOTO 3540 ELSE GOTO 2190
2190 IF SPACE(ACTIVITY,6)=2 AND SPACE(TRACK(PLACED,1),6)=1 THEN
      SFLAG=15:GOTO 3710 ELSE GOTO 2200
2200 IF SPACE(ACTIVITY,6)=2 AND SPACE(TRACK(PLACED,1),6)=3 THEN
      SFLAG=16:GOTO 3710 ELSE GOTO 2210
2210 STOP
2220 '
2230 'Square Courtyard Placement (North South Relationships)
2240 '
2250 ' North side check
2260 '
2270 FOR A=1 TO NPLACED
2280    IF TRACK(A,8)=YARDPTS(1,1) AND TRACK(A,9)=YARDPTS(1,2) THEN GOTO 2320
2290 NEXT A
2300 XDIST=YARDPTS(1,1)-PLANPTS(ACTIVITY,4,2)
2310 YDIST=YARDPTS(1,2)-PLANPTS(ACTIVITY,4,3):GOTO 670
2320 A=0:FLAG=1:SAVEPLACED=PLACED
```

```
2330 WHILE FLAG
2340    IF TRACK(PLACED,6)+NLENGTH>YARDPTS(2,1) THEN GOTO 2430
2350    IF TRACK(PLACED,9)<>YARDPTS(1,2) THEN GOTO 2430
2360    FOR A=1 TO NPLACED
2370      IF TRACK(A,9)<>YARDPTS(1,2) THEN GOTO 2400
2380      IF TRACK(A,8)>=TRACK(PLACED,6) AND TRACK(A,8)<=TRACK(PLACED,6)+NLENGTH
           THEN GOTO 2430
2390      IF TRACK(A,8)=TRACK(PLACED,6) AND TRACK(A,6)-TRACK(A,8)>NLENGTH
           THEN GOTO 2430
2400    NEXT A
2410    XDIST=TRACK(PLACED,6)-PLANPTS(ACTIVITY,4,2)
2420    YDIST=TRACK(PLACED,7)-PLANPTS(ACTIVITY,4,3):GOTO 670
2430    IF A=0 THEN A=1:PLACED=0 ELSE GOTO 2440
2440    PLACED=PLACED+1
2450    IF PLACED>NPLACED THEN FLAG=0
2460 WEND
2470 GOTO 3890
2480 '
2490 'South side check
2500 '
2510 FOR A=1 TO NPLACED
2520    IF TRACK(A,4)=YARDPTS(3,1) AND TRACK(A,5)=YARDPTS(3,2) THEN GOTO 2560
2530 NEXT A
2540 XDIST=YARDPTS(3,1)-PLANPTS(ACTIVITY,2,2)
2550 YDIST=YARDPTS(3,2)-PLANPTS(ACTIVITY,2,3):GOTO 670
2560 A=0:FLAG=1:SAVEPLACED=PLACED
2570 WHILE FLAG
2580    IF TRACK(PLACED,2)-SLENGTH<YARDPTS(4,1) THEN GOTO 2670
2590    IF TRACK(PLACED,3)<>YARDPTS(4,2) THEN GOTO 2670
2600    FOR B=1 TO NPLACED
2610      IF TRACK(B,3)<>YARDPTS(4,2) THEN GOTO 2640
2620      IF TRACK(B,4)<=TRACK(PLACED,2) AND TRACK(B,4)>=TRACK(PLACED,2)-SLENGTH
           THEN GOTO 2670
2630      IF TRACK(B,4)=TRACK(PLACED,2) AND TRACK(B,4)-TRACK(B,2)>SLENGTH
           THEN GOTO 2670
2640    NEXT B
2650    XDIST=TRACK(PLACED,2)-PLANPTS(ACTIVITY,2,2)
2660    YDIST=TRACK(PLACED,3)-PLANPTS(ACTIVITY,2,3):GOTO 670
2670    IF A=0 THEN A=1:PLACED=0 ELSE GOTO 2680
2680    PLACED=PLACED+1
2690    IF PLACED>NPLACED THEN FLAG=0
2700 WEND
2710 GOTO 3890
2720 '
2730 'North/South side check
2740 '
2750 FOR A=1 TO NPLACED
2760    IF TRACK(A,4)=YARDPTS(3,1) AND TRACK(A,5)=YARDPTS(3,2) THEN GOTO 2800
2770 NEXT A
2780 XDIST=YARDPTS(3,1)-PLANPTS(ACTIVITY,2,2)
2790 YDIST=YARDPTS(3,2)-PLANPTS(ACTIVITY,2,3):GOTO 670
2800 '
2810 'Check for A or B relationship with South placed activities
2820 '
2830 MAXREL=0
2840 FOR A=1 TO NPLACED
2850    IF TRACK(A,3)<>YARDPTS(4,2) THEN GOTO 2870
2860    IF TITLE(TRACK(A,1),ACTIVITY)>MAXREL THEN
         MAXREL=TITLE(TRACK(A,1),ACTIVITY):PLACED=A ELSE GOTO 2870
2870 NEXT A
```

```
2880 GOTO 2480
2890 '
2900 'South/North side check
2910 '
2920 FOR A=1 TO NPLACED
2930    IF TRACK(A,8)=YARDPTS(1,1) AND TRACK(A,9)=YARDPTS(1,2) THEN GOTO 2970
2940 NEXT A
2950 XDIST=YARDPTS(1,1)-PLANPTS(ACTIVITY,4,2)
2960 YDIST=YARDPTS(1,2)-PLANPTS(ACTIVITY,4,3):GOTO 670
2970 '
2980 'Check for A or B relationship with North placed activities
2990 '
3000 MAXREL=0
3010 FOR A=1 TO NPLACED
3020    IF TRACK(A,9)<>YARDPTS(1,2) THEN GOTO 3040
3030    IF TITLE(TRACK(A,1),ACTIVITY)>MAXREL THEN
         MAXREL=TITLE(TRACK(A,1),ACTIVITY):PLACED=A ELSE GOTO 3040
3040 NEXT A
3050 GOTO 2250
3060 '
3070 'East side check
3080 '
3090 FOR A=1 TO NPLACED
3100    IF TRACK(A,2)=YARDPTS(2,1) AND TRACK(A,3)=YARDPTS(2,2) THEN GOTO 3140
3110 NEXT A
3120 XDIST=YARDPTS(2,1)-PLANPTS(ACTIVITY,1,2)
3130 YDIST=YARDPTS(2,2)-PLANPTS(ACTIVITY,1,3):GOTO 670
3140 A=0:FLAG=1:SAVEPLACED=PLACED
3150 WHILE FLAG
3160    IF TRACK(PLACED,9)+ELENGTH>YARDPTS(3,2) THEN GOTO 3250
3170    IF TRACK(PLACED,2)<>YARDPTS(2,1) THEN GOTO 3250
3180    FOR C=1 TO NPLACED
3190      IF TRACK(C,2)<>YARDPTS(2,1) THEN GOTO 3220
3200      IF TRACK(C,3)>=TRACK(PLACED,9) AND TRACK(C,3)<=TRACK(PLACED,9)+ELENGTH
           THEN GOTO 3250
3210      IF TRACK(C,3)=TRACK(PLACED,9) AND TRACK(C,9)-TRACK(C,3)>ELENGTH
           THEN GOTO 3250
3220    NEXT C
3230    XDIST=TRACK(PLACED,8)-PLANPTS(ACTIVITY,1,2)
3240    YDIST=TRACK(PLACED,9)-PLANPTS(ACTIVITY,1,3):GOTO 670
3250    IF A=0 THEN A=1:PLACED=0 ELSE GOTO 3260
3260    PLACED=PLACED+1
3270    IF PLACED>NPLACED THEN FLAG=0
3280 WEND
3290 GOTO 3890
3300 '
3310 'West side check
3320 '
3330 FOR A=1 TO NPLACED
3340    IF TRACK(A,6)=YARDPTS(4,1) AND TRACK(A,7)=YARDPTS(4,2) THEN GOTO 3380
3350 NEXT A
3360 XDIST=YARDPTS(4,1)-PLANPTS(ACTIVITY,3,2)
3370 YDIST=YARDPTS(4,2)-PLANPTS(ACTIVITY,3,3):GOTO 670
3380 A=0:FLAG=1:SAVEPLACED=PLACED
3390 WHILE FLAG
3400    IF TRACK(PLACED,5)-WLENGTH<YARDPTS(1,2) THEN GOTO 3490
3410    IF TRACK(PLACED,4)<>YARDPTS(1,1) THEN GOTO 3490
3420    FOR D=1 TO NPLACED
3430      IF TRACK(D,4)<>YARDPTS(1,1) THEN GOTO 3460
3440      IF TRACK(D,7)<=TRACK(PLACED,5) AND TRACK(D,7)>=TRACK(PLACED,5)-WLENGTH
```

```
          THEN GOTO 3490
3450      IF TRACK(D,7)=TRACK(PLACED,5) AND TRACK(D,7)-TRACK(D,5)>WLENGTH
          THEN GOTO 3490
3460    NEXT D
3470    XDIST=TRACK(PLACED,4)-PLANPTS(ACTIVITY,3,2)
3480    YDIST=TRACK(PLACED,5)-PLANPTS(ACTIVITY,3,3):GOTO 670
3490    IF A=0 THEN A=1:PLACED=0 ELSE GOTO 3500
3500    PLACED=PLACED+1
3510    IF PLACED>NPLACED THEN FLAG=0
3520 WEND
3530 GOTO 3890
3540 '
3550 'East/West side check
3560 '
3570 FOR A=1 TO NPLACED
3580    IF TRACK(A,6)=YARDPTS(4,1) AND TRACK(A,7)=YARDPTS(4,2) THEN GOTO 3620
3590 NEXT A
3600 XDIST=YARDPTS(4,1)-PLANPTS(ACTIVITY,3,2)
3610 YDIST=YARDPTS(4,2)-PLANPTS(ACTIVITY,3,3):GOTO 670
3620 '
3630 'Check for highest priority with West placed activities
3640 '
3650 MAXREL=0
3660 FOR A=1 TO NPLACED
3670    IF TRACK(A,4)<>YARDPTS(4,1) THEN GOTO 3690
3680    IF TITLE(TRACK(A,1),ACTIVITY)>MAXREL THEN
         MAXREL=TITLE(TRACK(A,1),ACTIVITY):PLACED=A ELSE GOTO 3690
3690 NEXT A
3700 GOTO 3300
3710 '
3720 'West/East side check
3730 '
3740 FOR A=1 TO NPLACED
3750    IF TRACK(A,2)=YARDPTS(2,1) AND TRACK(A,3)=YARDPTS(2,2) THEN GOTO 3790
3760 NEXT A
3770 XDIST=YARDPTS(2,1)-PLANPTS(ACTIVITY,1,2)
3780 YDIST=YARDPTS(2,2)-PLANPTS(ACTIVITY,1,3):GOTO 670
3790 '
3800 'Check for highest proirity with East placed activities
3810 '
3820 MAXREL=0
3830 FOR A=1 TO NPLACED
3840    IF TRACK(A,2)<>YARDPTS(2,1) THEN GOTO 3860
3850    IF TITLE(TRACK(A,1),ACTIVITY)>MAXREL THEN
         MAXREL=TITLE(TRACK(A,1),ACTIVITY):PLACED=A ELSE GOTO 3860
3860 NEXT A
3870 GOTO 3060
3880 STOP
3890 '
3900 'No locations available on desired courtyard side
3910 '
3920 XDIST=250-PLANPTS(ACTIVITY,1,2)
3930 YDIST=150-PLANPTS(ACTIVITY,1,3)
3940 FOR A=1 TO SPACE(ACTIVITY,11)
3950    PLANPTS(ACTIVITY,A,2)=PLANPTS(ACTIVITY,A,2)+XDIST
3960    PLANPTS(ACTIVITY,A,3)=PLANPTS(ACTIVITY,A,3)+YDIST
3970 NEXT A
3980 FLAG=1:COUNT=0:I=1:PLACED=SAVEPLACED
3990 WHILE FLAG
4000    I=PLANPTS(ACTIVITY,I,1)
```

```
4010    LINE (PLANPTS(ACTIVITY,PLANPTS(ACTIVITY,I,1),2),PLANPTS(ACTIVITY,
          PLANPTS(ACTIVITY,I,1),3))-(PLANPTS(ACTIVITY,PLANPTS(ACTIVITY,I,4)
          ,2),PLANPTS(ACTIVITY,PLANPTS(ACTIVITY,I,4),3)),1
4020    I=PLANPTS(ACTIVITY,I,4)
4030    COUNT=COUNT+1
4040    IF COUNT>=SPACE(ACTIVITY,11) THEN FLAG=0
4050 WEND
4060 LOCATE 21,1:PRINT USING "#";ACTIVITY
4070 LOCATE 21,4:PRINT TITLE$(ACTIVITY,ACTIVITY)
4080 LOCATE 22,1:PRINT TITLE$(ACTIVITY,TRACK(PLACED,1))
4090 LOCATE 22,2:PRINT "-RELATIONSHIP WITH ";TITLE$(TRACK(PLACED,1),
     TRACK(PLACED,1))
4100 LOCATE 23,1:PRINT "LOCATE ON COURTYARD"
4110 FOR A=1 TO 500:NEXT A
4120 PEN ON
4130 IF PEN(3)=0 THEN GOTO 4120
4140 X1=PEN(4):Y1=PEN(5)
4150 PSET (X1,Y1)
4160 LOCATE 21,1:PRINT SPC(39)
4170 LOCATE 22,1:PRINT SPC(39)
4180 LOCATE 23,1:PRINT "CORRECT POINT?      YES    NO      "
4190 LINE (150,185)-(175,170),1,B:LINE (195,185)-(220,170),1,B
4200 PEN ON
4210 IF PEN(3)=0 THEN GOTO 4200
4220 IF PEN(8)>21 AND PEN(9)>24 THEN PSET (X1,Y1),0:GOTO 4110 ELSE GOTO 4230
4230 PSET (X1,Y1),0
4240 IF X1>=YARDPTS(1,1)-8 AND X1<=YARDPTS(1,1) AND Y1>=YARDPTS(1,2) AND
     Y1<=YARDPTS(4,2) THEN XDIST=YARDPTS(1,1)-PLANPTS(ACTIVITY,2,2):
     YDIST=Y1-PLANPTS(ACTIVITY,2,3):SPACE(ACTIVITY,6)=4:GOTO 4290 ELSE GOTO 4250

4250 IF X1>=YARDPTS(2,1) AND X1<=YARDPTS(2,1)+8 AND Y1>=YARDPTS(2,2) AND
     Y1<=YARDPTS(3,2) THEN XDIST=YARDPTS(2,1)-PLANPTS(ACTIVITY,1,2):
     YDIST=Y1-PLANPTS(ACTIVITY,1,3):SPACE(ACTIVITY,6)=2:GOTO 4290 ELSE GOTO 4260

4260 IF Y1>=YARDPTS(1,2)-8 AND Y1<=YARDPTS(1,2) AND X1>=YARDPTS(1,1) AND
     X1<=YARDPTS(2,1) THEN XDIST=X1-PLANPTS(ACTIVITY,4,2):YDIST=YARDPTS(1,2)-
     PLANPTS(ACTIVITY,4,3):SPACE(ACTIVITY,6)=1:GOTO 4290 ELSE GOTO 4270
4270 IF Y1>=YARDPTS(4,2) AND Y1<=YARDPTS(4,2)+8 AND X1>=YARDPTS(4,1) AND
     X1<=YARDPTS(3,1) THEN XDIST=X1-PLANPTS(ACTIVITY,1,2):YDIST=YARDPTS(4,2)-
     PLANPTS(ACTIVITY,1,3):SPACE(ACTIVITY,6)=3:GOTO 4290 ELSE STOP
4280 '
4290 FLAG=1:COUNT=0:I=1
4300 WHILE FLAG
4310    I=PLANPTS(ACTIVITY,I,1)
4320    LINE (PLANPTS(ACTIVITY,PLANPTS(ACTIVITY,I,1),2),PLANPTS(ACTIVITY,
          PLANPTS(ACTIVITY,I,1),3))-(PLANPTS(ACTIVITY,PLANPTS(ACTIVITY,I,4)
          ,2),PLANPTS(ACTIVITY,PLANPTS(ACTIVITY,I,4),3)),0
4330    I=PLANPTS(ACTIVITY,I,4)
4340    COUNT=COUNT+1
4350    IF COUNT>=SPACE(ACTIVITY,11) THEN FLAG=0
4360 WEND
4370 IFLAG=1:GOTO 670
4380 '
4390 'Modify location of activity
4400 '
4410 LOCATE 23,1:PRINT "ANY CHANGES?        YES    NO      "
4420 LINE (150,185)-(175,170),1,B:LINE (195,185)-(220,170),1,B
4430 PEN ON
4440 IF PEN(3)=0 THEN GOTO 4430
4450 IF PEN(8)>21 AND PEN(9)>24 THEN GOTO 1440
```

```
4460 LINE (150,185)-(175,170),0,B:LINE (195,185)-(220,170),0,B
4470 LOCATE 23,1:PRINT "  MODIFY   REDRAW                "
4480 LINE (5,185)-(75,170),1,B:LINE (80,185)-(150,170),1,B
4490 PEN ON
4500 IF PEN(3)=0 THEN GOTO 4490
4510 IF PEN(8)>21 AND PEN(9)>10 THEN GOTO 5250
4520 LINE (5,185)-(75,170),0,B:LINE (80,185)-(150,170),0,B
4530 LOCATE 23,1:PRINT SPC(39)
4540 LOCATE 21,1:PRINT "INDICATE NEW ACTIVITY LOCATION"
4550 FOR B=1 TO 500:NEXT B
4560 PEN ON
4570 IF PEN(3)=0 THEN GOTO 4560
4580 X1=PEN(4):Y1=PEN(5)
4590 PSET (X1,Y1)
4600 LOCATE 23,1:PRINT "CORRECT POINT?     YES   NO       "
4610 LINE (150,185)-(175,170),1,B:LINE (195,185)-(220,170),1,B
4620 PEN ON
4630 IF PEN(3)=0 THEN GOTO 4620
4640 IF PEN(8)>21 AND PEN(9)>24 THEN PSET (X1,Y1),0:GOTO 4550 ELSE GOTO 4650
4650 PSET (X1,Y1),0
4660 '
4670 'Check TRACK array to determine activity to be moved
4680 '
4690 FOR A=1 TO N
4700    IF X1>=PLANPTS(A,1,2) AND X1<=PLANPTS(A,2,2) AND Y1>=PLANPTS(A,1,3) AND
         Y1<=PLANPTS(A,4,3) THEN ACTIVITY=A:GOTO 4730 ELSE GOTO 4710
4710 NEXT A
4720 STOP
4730 FOR A=1 TO NPLACED
4740    IF ACTIVITY=TRACK(A,1) THEN GOTO 4770
4750 NEXT A
4760 STOP
4770 LOCATE 1,38:PRINT "*"
4780 FOR B=1 TO 500:NEXT B
4790 PEN ON
4800 IF PEN(3)=0 THEN GOTO 4790
4810 X2=PEN(4):Y2=PEN(5)
4820 PSET (X2,Y2)
4830 FOR B=1 TO 500:NEXT B
4840 PEN ON
4850 IF PEN(3)=0 THEN GOTO 4840
4860 IF PEN(8)>21 AND PEN(9)>24 THEN PSET (X2,Y2),0:GOTO 4780 ELSE GOTO 4870
4870 PSET (X2,Y2),0
4880 LOCATE 1,38:PRINT "  "
4890 '
4900 'West side courtyard check
4910 '
4920 IF X2>=YARDPTS(1,1)-8 AND X2<=YARDPTS(1,1) AND Y2>=YARDPTS(1,2) AND
     Y2<=YARDPTS(4,2) THEN XDIST=YARDPTS(1,1)-PLANPTS(ACTIVITY,2,2):
     YDIST=Y2-PLANPTS(ACTIVITY,2,3):SPACE(ACTIVITY,6)=4:GOTO 5060 ELSE GOTO 4930

4930 '
4940 'East side courtyard check
4950 '
4960 IF X2>=YARDPTS(2,1) AND X2<=YARDPTS(2,1)+8 AND Y2>=YARDPTS(2,2) AND
     Y2<=YARDPTS(3,2) THEN XDIST=YARDPTS(2,1)-PLANPTS(ACTIVITY,1,2):
     YDIST=Y2-PLANPTS(ACTIVITY,1,3):SPACE(ACTIVITY,6)=2:GOTO 5060 ELSE GOTO 4970

4970 '
4980 'North side courtyard check
```

```
4990 '
5000 IF Y2>=YARDPTS(1,2)-8 AND Y2<=YARDPTS(1,2) AND X2>=YARDPTS(1,1) AND
     X2<=YARDPTS(2,1) THEN XDIST=X2-PLANPTS(ACTIVITY,4,2):YDIST=YARDPTS(1,2)-
     PLANPTS(ACTIVITY,4,3):SPACE(ACTIVITY,6)=1:GOTO 5060 ELSE GOTO 5010
5010 '
5020 'South side courtyard check
5030 '
5040 IF Y2>=YARDPTS(4,2) AND Y2<=YARDPTS(4,2)+8 AND X2>=YARDPTS(4,1) AND
     X2<=YARDPTS(3,1) THEN XDIST=X2-PLANPTS(ACTIVITY,1,2):YDIST=YARDPTS(4,2)-
     PLANPTS(ACTIVITY,1,3):SPACE(ACTIVITY,6)=3:GOTO 5060 ELSE GOTO 5050
5050 STOP
5060 '
5070 'Erase existing activity
5080 '
5090 XPLACE=INT(PLANPTS(ACTIVITY,1,2)/8):YPLACE=INT(PLANPTS(ACTIVITY,1,3)/8)
5100 IF TRACK(A,7)=YARDPTS(1,2) AND TRACK(A,9)=YARDPTS(1,2) THEN
      LOCATE YPLACE,XPLACE+1:GOTO 5140 ELSE GOTO 5110
5110 IF TRACK(A,2)=YARDPTS(2,1) AND TRACK(A,8)=YARDPTS(2,1) THEN
      LOCATE YPLACE,XPLACE+5:GOTO 5140 ELSE GOTO 5120
5120 IF TRACK(A,3)=YARDPTS(3,2) AND TRACK(A,5)=YARDPTS(3,2) THEN
      LOCATE YPLACE+5,XPLACE+1:GOTO 5140 ELSE GOTO 5130
5130 IF TRACK(A,4)=YARDPTS(4,1) AND TRACK(A,6)=YARDPTS(4,1) THEN
      LOCATE YPLACE,XPLACE+1:GOTO 5140 ELSE STOP
5140 IF ACTIVITY< 10 THEN PRINT " "
5150 IF ACTIVITY>=10 THEN PRINT "  "
5160 FLAG=1:COUNT=0:I=1
5170 WHILE FLAG
5180    I=PLANPTS(ACTIVITY,I,1)
5190    LINE (PLANPTS(ACTIVITY,PLANPTS(ACTIVITY,I,1),2),PLANPTS(ACTIVITY,
         PLANPTS(ACTIVITY,I,1),3))-(PLANPTS(ACTIVITY,PLANPTS(ACTIVITY,I,4),2),
         PLANPTS(ACTIVITY,PLANPTS(ACTIVITY,I,4),3)),0
5200    I=PLANPTS(ACTIVITY,I,4)
5210    COUNT=COUNT+1
5220    IF COUNT>=SPACE(ACTIVITY,11) THEN FLAG=0
5230 WEND
5240 CFLAG=1:GOTO 670
5250 '
5260 'Redraw screen
5270 '
5280 LINE (5,185)-(75,170),0,B:LINE (80,185)-(150,170),0,B
5290 LOCATE 23,1:PRINT SPC(39)
5300 FOR A=1 TO NPLACED
5310 FLAG=1:COUNT=0:I=1
5320 WHILE FLAG
5330    I=PLANPTS(TRACK(A,1),I,1)
5340    LINE (PLANPTS(TRACK(A,1),PLANPTS(TRACK(A,1),I,1),2),PLANPTS(TRACK(A,1),
        PLANPTS(TRACK(A,1),I,1),3))-(PLANPTS(TRACK(A,1),PLANPTS(TRACK(A,1),I,4)
        ,2),PLANPTS(TRACK(A,1),PLANPTS(TRACK(A,1),I,4),3)),1
5350    I=PLANPTS(TRACK(A,1),I,4)
5360    COUNT=COUNT+1
5370    IF COUNT>=SPACE(TRACK(A,1),11) THEN FLAG=0
5380 WEND
5390 NEXT A
5400 LINE (YARDPTS(1,1),YARDPTS(1,2))-(YARDPTS(3,1),YARDPTS(3,2)),1,B
5410 GOTO 4380
5420 '
5430 'Allow another circulation design to be developed
5440 '
5450 LOCATE 21,1:PRINT SPC(39)
5460 LINE (150,185)-(175,170),0,B:LINE (195,185)-(220,170),0,B
```

```
5470 LOCATE 23,1:PRINT SPC(39)
5480 LOCATE 23,1:PRINT "SAVE PLAN?          YES   NO     "
5490 LINE (150,185)-(175,170),1,B:LINE (195,185)-(220,170),1,B
5500 PEN ON
5510 IF PEN(3)=0 THEN GOTO 5500
5520 IF PEN(8)>21 AND PEN(9)<23 THEN JUMP$="POINT":GOTO 5780 ELSE GOTO 5530
5530 LOCATE 23,1:PRINT "PLACE ON SITE?      YES   NO     "
5540 LINE (150,185)-(175,170),1,B:LINE (195,185)-(220,170),1,B
5550 PEN ON
5560 IF PEN(3)=0 THEN GOTO 5550
5570 IF PEN(8)>21 AND PEN(9)<23 THEN GOTO 5770
5580 LOCATE 23,1:PRINT "ANOTHER DESIGN?     YES   NO     "
5590 LINE (150,185)-(175,170),1,B:LINE (195,185)-(220,170),1,B
5600 PEN ON
5610 IF PEN(3)=0 THEN GOTO 5600
5620 IF PEN(8)>21 AND PEN(9)>24 THEN GOTO 5690
5630 LOCATE 23,1:PRINT "UPDATE DATA?        YES   NO     "
5640 LINE (150,185)-(175,170),1,B:LINE (195,185)-(220,170),1,B
5650 PEN ON
5660 IF PEN(3)=0 THEN GOTO 5650
5670 IF PEN(8)>21 AND PEN(9)<23 THEN GOTO 5750
5680 GOTO 5760
5690 LOCATE 23,1:PRINT "START AGAIN?        YES   NO     "
5700 LINE (150,185)-(175,170),1,B:LINE (195,185)-(220,170),1,B
5710 PEN ON
5720 IF PEN(3)=0 THEN GOTO 5710
5730 IF PEN(8)>21 AND PEN(9)<23 THEN GOTO 5740 ELSE STOP
5740 CHAIN MERGE "CREATE",10,DELETE 10-5730
5750 CHAIN MERGE "SPACE",10,ALL,DELETE 10-5740
5760 CHAIN MERGE "GEOMTRY",10,ALL,DELETE 10-5750
5770 CHAIN MERGE "SITE",10,ALL,DELETE 10-5760
5780 CHAIN MERGE "PLNFILE",10,ALL,DELETE 10-5770
```

```
10 '*************************************************************************
20 'RADIAL Subroutine
30 '*************************************************************************
40 'This routine orders activities around a pinwheel shaped circulation
50 'pattern.  The user enters courtyard dimensions, hall widths and
60 'hall length.  Activities are placed around and within pattern.
70 CLS:PEN OFF
80 FOR A=1 TO 4
90      CORNER(A)=0
100 NEXT A
110 CFLAG=0:JUMP=3
120 LOCATE 2,13:PRINT "RADIAL Subroutine"
130 LOCATE 8,1 :PRINT "  PINWHEEL"
140 LOCATE 14,1:PRINT "  CROSS GRID"
150 LINE (10,50)-(100,65),1,B:LINE (10,100)-(100,115),1,B
160 LINE (125,55)-(135,65),1,B
170 LINE (125,65)-(122,50),1,B:LINE (125,55)-(140,52),1,B:
    LINE (135,55)-(138,70),1,B:LINE (135,65)-(120,68),1,B
180 LINE (125,105)-(135,115),1,B
190 LINE (122,100)-(125,120),1,B:LINE (135,100)-(138,120),1,B
200 LINE (120,102)-(140,105),1,B:LINE (120,115)-(140,118),1,B
210 PEN ON
220 IF PEN(3)=0 THEN GOTO 210
230 IF PEN(8)>6 AND PEN(8)<12 AND PEN(9)<15 THEN GOTO 260
240 IF PEN(8)>11 AND PEN(8)<15 AND PEN(9)<15 THEN GOTO 250
250 STOP
260 '
270 'Pinwheel placement
280 '
290 LOCATE 21,1:INPUT "ENTER COURTYARD DIMENSION  ",CENTER
300 LOCATE 22,1:INPUT "ENTER AISLE WIDTH          ",WHALL
310 LOCATE 23,1:INPUT "ENTER AISLE LENGTH         ",LHALL
320 CENTER=CENTER*SCALE:WHALL=WHALL*SCALE:LHALL=LHALL*SCALE
330 CLS
340 LOCATE 23,1:PRINT "INTERACTIVE ?      YES   NO"
350 LINE (150,185)-(175,170),1,B:LINE (195,185)-(220,170),1,B
360 PEN ON
370 IF PEN(3)=0 THEN GOTO 360
380 IF PEN(8)>21 AND PEN(9)>24 THEN IFLAG=0 ELSE IFLAG=1
390 CLS
400 '
410 'Draw skeleton pinwheel
420 '
430 PINPTS(1,1)=160-CENTER/2:PINPTS(1,2)=50
440 PINPTS(2,1)=160+CENTER/2:PINPTS(2,2)=50
450 PINPTS(3,1)=160+CENTER/2:PINPTS(3,2)=50+CENTER
460 PINPTS(4,1)=160-CENTER/2:PINPTS(4,2)=50+CENTER
470 HALLPTS(1,1)=PINPTS(1,1)-WHALL:HALLPTS(1,2)=PINPTS(1,2)-LHALL
480 HALLPTS(2,1)=PINPTS(2,1)+LHALL:HALLPTS(2,2)=PINPTS(2,2)-WHALL
490 HALLPTS(3,1)=PINPTS(3,1)+WHALL:HALLPTS(3,2)=PINPTS(3,2)+LHALL
500 HALLPTS(4,1)=PINPTS(4,1)-LHALL:HALLPTS(4,2)=PINPTS(4,2)+WHALL
510 HALLPTS(5,1)=PINPTS(1,1):HALLPTS(5,2)=HALLPTS(1,2)
520 HALLPTS(6,1)=HALLPTS(2,1):HALLPTS(6,2)=PINPTS(2,2)
530 HALLPTS(7,1)=PINPTS(3,1):HALLPTS(7,2)=HALLPTS(3,2)
540 HALLPTS(8,1)=HALLPTS(4,1):HALLPTS(8,2)=PINPTS(4,2)
550 HALLPTS(9,1)=PINPTS(1,1):HALLPTS(9,2)=HALLPTS(2,2)
560 HALLPTS(10,1)=HALLPTS(3,1):HALLPTS(10,2)=PINPTS(2,2)
570 HALLPTS(11,1)=PINPTS(3,1):HALLPTS(11,2)=HALLPTS(4,2)
580 HALLPTS(12,1)=HALLPTS(1,1):HALLPTS(12,2)=PINPTS(4,2)
590 LINE (PINPTS(1,1),PINPTS(1,2))-(PINPTS(3,1),PINPTS(3,2)),1,B
```

```
600 LINE (HALLPTS(1,1),HALLPTS(1,2))-(PINPTS(4,1),PINPTS(4,2)),1,B
610 LINE (HALLPTS(2,1),HALLPTS(2,2))-(PINPTS(1,1),PINPTS(1,2)),1,B
620 LINE (HALLPTS(3,1),HALLPTS(3,2))-(PINPTS(2,1),PINPTS(2,2)),1,B
630 LINE (HALLPTS(4,1),HALLPTS(4,2))-(PINPTS(3,1),PINPTS(3,2)),1,B
640 '
650 'Determine activity with largest square footage
660 '
670 MAXAREA=0:NPLACED=0
680 FOR A=1 TO N
690     IF SPACE(A,1)>MAXAREA THEN ACTIVITY=A:MAXAREA=SPACE(A,1) ELSE GOTO 700
700 NEXT A
710 SIDE=ABS(PLANPTS(ACTIVITY,2,3)-PLANPTS(ACTIVITY,3,3))
720 LENGTH=ABS(PLANPTS(ACTIVITY,1,2)-PLANPTS(ACTIVITY,2,2))
730 IF SIDE>CENTER OR LENGTH>CENTER THEN GOTO 900
740 '
750 'Check wall assignment and determine initial activity placement
760 '
770 IF SPACE(ACTIVITY,6)=4 THEN GOTO 870
780 IF SPACE(ACTIVITY,6)=3 THEN GOTO 850
790 IF SPACE(ACTIVITY,6)=2 THEN GOTO 830
800 IF SPACE(ACTIVITY,6)=0 THEN SPACE(ACTIVITY,6)=1
810 XDIST=PINPTS(1,1)-PLANPTS(ACTIVITY,1,2)
820 YDIST=PINPTS(1,2)-PLANPTS(ACTIVITY,1,3):GOTO 1020
830 XDIST=PINPTS(2,1)-PLANPTS(ACTIVITY,2,2)
840 YDIST=PINPTS(2,2)-PLANPTS(ACTIVITY,2,3):GOTO 1020
850 XDIST=PINPTS(3,1)-PLANPTS(ACTIVITY,3,2)
860 YDIST=PINPTS(3,2)-PLANPTS(ACTIVITY,3,3):GOTO 1020
870 XDIST=PINPTS(4,1)-PLANPTS(ACTIVITY,4,2)
880 YDIST=PINPTS(4,2)-PLANPTS(ACTIVITY,4,3):GOTO 1020
890 '
900 IF SPACE(ACTIVITY,6)=4 THEN GOTO 1000
910 IF SPACE(ACTIVITY,6)=3 THEN GOTO 980
920 IF SPACE(ACTIVITY,6)=2 THEN GOTO 960
930 IF SPACE(ACTIVITY,6)=0 THEN SPACE(ACTIVITY,6)=1
940 XDIST=PINPTS(1,1)-PLANPTS(ACTIVITY,4,2)
950 YDIST=PINPTS(1,2)-PLANPTS(ACTIVITY,4,3)-WHALL:GOTO 1020
960 XDIST=PINPTS(2,1)-PLANPTS(ACTIVITY,1,2)+WHALL
970 YDIST=PINPTS(2,2)-PLANPTS(ACTIVITY,1,3):GOTO 1020
980 XDIST=PINPTS(3,1)-PLANPTS(ACTIVITY,2,2)
990 YDIST=PINPTS(3,2)-PLANPTS(ACTIVITY,2,3)+WHALL:GOTO 1020
1000 XDIST=PINPTS(4,1)-PLANPTS(ACTIVITY,3,2)-WHALL
1010 YDIST=PINPTS(4,2)-PLANPTS(ACTIVITY,3,3):GOTO 1020
1020 '
1030 'Translate activity coordinates to pinwheel
1040 '
1050 FOR A=1 TO SPACE(ACTIVITY,11)
1060     PLANPTS(ACTIVITY,A,2)=PLANPTS(ACTIVITY,A,2)+XDIST
1070     PLANPTS(ACTIVITY,A,3)=PLANPTS(ACTIVITY,A,3)+YDIST
1080 NEXT A
1090 '
1100 IF PLANPTS(ACTIVITY,3,2)>PINPTS(2,1) AND PLANPTS(ACTIVITY,3,3)=PINPTS(2,2)
      THEN CORNER(2)=1
1110 IF PLANPTS(ACTIVITY,4,2)=PINPTS(3,1) AND PLANPTS(ACTIVITY,4,3)>PINPTS(3,2)
      THEN CORNER(3)=1
1120 IF PLANPTS(ACTIVITY,1,2)<PINPTS(4,1) AND PLANPTS(ACTIVITY,1,3)=PINPTS(4,2)
      THEN CORNER(4)=1
1130 IF PLANPTS(ACTIVITY,2,2)=PINPTS(1,1) AND PLANPTS(ACTIVITY,2,3)<PINPTS(1,2)
      THEN CORNER(1)=1
1140 '
1150 '
```

```
1160 'Draw activity on pinwheel
1170 '
1180 FLAG=1:I=1:COUNT=0
1190 IF CFLAG=0 THEN NPLACED=NPLACED+1
1200 WHILE FLAG
1210     I=PLANPTS(ACTIVITY,I,1)
1220     LINE (PLANPTS(ACTIVITY,PLANPTS(ACTIVITY,I,1),2),PLANPTS(ACTIVITY,
           PLANPTS(ACTIVITY,I,1),3))-(PLANPTS(ACTIVITY,PLANPTS(ACTIVITY,I,4),2)
           ,PLANPTS(ACTIVITY,PLANPTS(ACTIVITY,I,4),3)),1
1230     I=PLANPTS(ACTIVITY,I,4)
1240     COUNT=COUNT+1
1250     IF COUNT>=SPACE(ACTIVITY,11) THEN FLAG=0
1260 WEND
1270 '
1280 'Store reference points located on pinwheel
1290 '
1300 IF CFLAG=0 THEN GOTO 1350
1310 FOR A=1 TO NPLACED
1320    IF ACTIVITY=TRACK(A,1) THEN M=A:GOTO 1370 ELSE GOTO 1330
1330 NEXT A
1340 STOP
1350 M=NPLACED
1360 TRACK(M,1)=ACTIVITY
1370 IF PLANPTS(ACTIVITY,1,3)=PINPTS(1,2) AND PLANPTS(ACTIVITY,2,3)=50 AND
      PLANPTS(ACTIVITY,1,2)=PINPTS(1,1) THEN GOTO 1380 ELSE GOTO 1420
1380 TRACK(M,2)=PLANPTS(ACTIVITY,1,2):TRACK(M,3)=PLANPTS(ACTIVITY,1,3)
1390 TRACK(M,4)=PLANPTS(ACTIVITY,2,2):TRACK(M,5)=PLANPTS(ACTIVITY,2,3)
1400 TRACK(M,6)=0                     :TRACK(M,7)=0
1410 TRACK(M,8)=PLANPTS(ACTIVITY,4,2):TRACK(M,9)=PLANPTS(ACTIVITY,4,3):GOTO 2220

1420 IF PLANPTS(ACTIVITY,2,2)=PINPTS(2,1) AND PLANPTS(ACTIVITY,3,2)=160+CENTER/2
      AND PLANPTS(ACTIVITY,2,3)=PINPTS(2,2) THEN GOTO 1430 ELSE GOTO 1480
1430 TRACK(M,2)=PLANPTS(ACTIVITY,1,2):TRACK(M,3)=PLANPTS(ACTIVITY,1,3)
1440 TRACK(M,4)=PLANPTS(ACTIVITY,2,2):TRACK(M,5)=PLANPTS(ACTIVITY,2,3)
1450 TRACK(M,6)=PLANPTS(ACTIVITY,3,2):TRACK(M,7)=PLANPTS(ACTIVITY,3,3)
1460 TRACK(M,8)=0                     :TRACK(M,9)=0:GOTO 2220
1470 '
1480 IF PLANPTS(ACTIVITY,3,3)=PINPTS(3,2) AND PLANPTS(ACTIVITY,4,3)=50+CENTER
      AND PLANPTS(ACTIVITY,3,2)=PINPTS(3,1) THEN GOTO 1490 ELSE GOTO 1530
1490 TRACK(M,2)=0                     :TRACK(M,3)=0
1500 TRACK(M,4)=PLANPTS(ACTIVITY,2,2):TRACK(M,5)=PLANPTS(ACTIVITY,2,3)
1510 TRACK(M,6)=PLANPTS(ACTIVITY,3,2):TRACK(M,7)=PLANPTS(ACTIVITY,3,3)
1520 TRACK(M,8)=PLANPTS(ACTIVITY,4,2):TRACK(M,9)=PLANPTS(ACTIVITY,4,3):GOTO 2220

1530 IF PLANPTS(ACTIVITY,4,2)=PINPTS(4,1) AND PLANPTS(ACTIVITY,1,2)=160-CENTER/2
      AND PLANPTS(ACTIVITY,4,3)=PINPTS(4,2) THEN GOTO 1540 ELSE GOTO 1580
1540 TRACK(M,2)=PLANPTS(ACTIVITY,1,2):TRACK(M,3)=PLANPTS(ACTIVITY,1,3)
1550 TRACK(M,4)=0                     :TRACK(M,5)=0
1560 TRACK(M,6)=PLANPTS(ACTIVITY,3,2):TRACK(M,7)=PLANPTS(ACTIVITY,3,3)
1570 TRACK(M,8)=PLANPTS(ACTIVITY,4,2):TRACK(M,9)=PLANPTS(ACTIVITY,4,3):GOTO 2220

1580 IF PLANPTS(ACTIVITY,4,3)=50-WHALL AND PLANPTS(ACTIVITY,3,3)=50-WHALL AND
      PLANPTS(ACTIVITY,4,2)=PINPTS(1,1) THEN GOTO 1590 ELSE GOTO 1630
1590 TRACK(M,2)=PLANPTS(ACTIVITY,1,2):TRACK(M,3)=PLANPTS(ACTIVITY,1,3)
1600 TRACK(M,4)=0                     :TRACK(M,5)=0
1610 TRACK(M,6)=PLANPTS(ACTIVITY,3,2):TRACK(M,7)=PLANPTS(ACTIVITY,3,3)
1620 TRACK(M,8)=PLANPTS(ACTIVITY,4,2):TRACK(M,9)=PLANPTS(ACTIVITY,4,3):GOTO 2220

1630 IF PLANPTS(ACTIVITY,1,2)=PINPTS(2,1)+WHALL AND PLANPTS(ACTIVITY,4,2)
      =PINPTS(2,1)+WHALL AND PLANPTS(ACTIVITY,1,3)=PINPTS(2,2) THEN
```

```
     GOTO 1640 ELSE GOTO 1690
1640 TRACK(M,2)=PLANPTS(ACTIVITY,1,2):TRACK(M,3)=PLANPTS(ACTIVITY,1,3)
1650 TRACK(M,4)=PLANPTS(ACTIVITY,2,2):TRACK(M,5)=PLANPTS(ACTIVITY,2,3)
1660 TRACK(M,6)=0                     :TRACK(M,7)=0
1670 TRACK(M,8)=PLANPTS(ACTIVITY,4,2):TRACK(M,9)=PLANPTS(ACTIVITY,4,3):GOTO 2220

1680 '
1690 IF PLANPTS(ACTIVITY,1,3)=PINPTS(3,2)+WHALL AND PLANPTS(ACTIVITY,2,3)
     =PINPTS(3,2)+WHALL AND PLANPTS(ACTIVITY,2,2)=PINPTS(3,1) THEN
     GOTO 1700 ELSE GOTO 1740
1700 TRACK(M,2)=PLANPTS(ACTIVITY,1,2):TRACK(M,3)=PLANPTS(ACTIVITY,1,3)
1710 TRACK(M,4)=PLANPTS(ACTIVITY,2,2):TRACK(M,5)=PLANPTS(ACTIVITY,2,3)
1720 TRACK(M,6)=PLANPTS(ACTIVITY,3,2):TRACK(M,7)=PLANPTS(ACTIVITY,3,3)
1730 TRACK(M,8)=0                     :TRACK(M,9)=0:GOTO 2220
1740 IF PLANPTS(ACTIVITY,3,2)=PINPTS(4,1)-WHALL AND PLANPTS(ACTIVITY,2,2)
     =160-CENTER/2-WHALL AND PLANPTS(ACTIVITY,3,3)=PINPTS(4,2) THEN GOTO 1750
     ELSE GOTO 1790
1750 TRACK(M,2)=0                     :TRACK(M,3)=0
1760 TRACK(M,4)=PLANPTS(ACTIVITY,2,2):TRACK(M,5)=PLANPTS(ACTIVITY,2,3)
1770 TRACK(M,6)=PLANPTS(ACTIVITY,3,2):TRACK(M,7)=PLANPTS(ACTIVITY,3,3)
1780 TRACK(M,8)=PLANPTS(ACTIVITY,4,2):TRACK(M,9)=PLANPTS(ACTIVITY,4,3):GOTO 2220

1790 IF PLANPTS(ACTIVITY,1,3)=PINPTS(1,2) AND PLANPTS(ACTIVITY,2,3)=50
     THEN GOTO 1800 ELSE GOTO 1840
1800 TRACK(M,2)=PLANPTS(ACTIVITY,1,2):TRACK(M,3)=PLANPTS(ACTIVITY,1,3)
1810 TRACK(M,4)=PLANPTS(ACTIVITY,2,2):TRACK(M,5)=PLANPTS(ACTIVITY,2,3)
1820 TRACK(M,6)=0                     :TRACK(M,7)=0
1830 TRACK(M,8)=0                     :TRACK(M,9)=0:GOTO 2220
1840 IF PLANPTS(ACTIVITY,2,2)=PINPTS(2,1) AND PLANPTS(ACTIVITY,3,2)=160+CENTER/2
     THEN GOTO 1850 ELSE GOTO 1900
1850 TRACK(M,2)=0                     :TRACK(M,3)=0
1860 TRACK(M,4)=PLANPTS(ACTIVITY,2,2):TRACK(M,5)=PLANPTS(ACTIVITY,2,3)
1870 TRACK(M,6)=PLANPTS(ACTIVITY,3,2):TRACK(M,7)=PLANPTS(ACTIVITY,3,3)
1880 TRACK(M,8)=0                     :TRACK(M,9)=0:GOTO 2220
1890 '
1900 IF PLANPTS(ACTIVITY,3,3)=PINPTS(3,2) AND PLANPTS(ACTIVITY,4,3)=50+CENTER
       THEN GOTO 1910 ELSE GOTO 1960
1910 TRACK(M,2)=0                     :TRACK(M,3)=0
1920 TRACK(M,4)=0                     :TRACK(M,5)=0
1930 TRACK(M,6)=PLANPTS(ACTIVITY,3,2):TRACK(M,7)=PLANPTS(ACTIVITY,3,3)
1940 TRACK(M,8)=PLANPTS(ACTIVITY,4,2):TRACK(M,9)=PLANPTS(ACTIVITY,4,3):GOTO 2220

1950 '
1960 IF PLANPTS(ACTIVITY,4,2)=PINPTS(4,1) AND PLANPTS(ACTIVITY,1,2)=160-CENTER/2
     THEN GOTO 1970 ELSE GOTO 2010
1970 TRACK(M,2)=PLANPTS(ACTIVITY,1,2):TRACK(M,3)=PLANPTS(ACTIVITY,1,3)
1980 TRACK(M,4)=0                     :TRACK(M,5)=0
1990 TRACK(M,6)=0                     :TRACK(M,7)=0
2000 TRACK(M,8)=PLANPTS(ACTIVITY,4,2):TRACK(M,9)=PLANPTS(ACTIVITY,4,3):GOTO 2220

2010 IF PLANPTS(ACTIVITY,4,3)=50-WHALL AND PLANPTS(ACTIVITY,3,3)=50-WHALL
     THEN GOTO 2020 ELSE GOTO 2060
2020 TRACK(M,2)=0                     :TRACK(M,3)=0
2030 TRACK(M,4)=0                     :TRACK(M,5)=0
2040 TRACK(M,6)=PLANPTS(ACTIVITY,3,2):TRACK(M,7)=PLANPTS(ACTIVITY,3,3)
2050 TRACK(M,8)=PLANPTS(ACTIVITY,4,2):TRACK(M,9)=PLANPTS(ACTIVITY,4,3):GOTO 2220

2060 IF PLANPTS(ACTIVITY,1,2)=PINPTS(2,1)+WHALL AND PLANPTS(ACTIVITY,4,2)=
     PINPTS(2,1)+WHALL THEN GOTO 2070 ELSE GOTO 2110
2070 TRACK(M,2)=PLANPTS(ACTIVITY,1,2):TRACK(M,3)=PLANPTS(ACTIVITY,1,3)
```

```
2080 TRACK(M,4)=0                        :TRACK(M,5)=0
2090 TRACK(M,6)=0                        :TRACK(M,7)=0
2100 TRACK(M,8)=PLANPTS(ACTIVITY,4,2):TRACK(M,9)=PLANPTS(ACTIVITY,4,3):GOTO 2220

2110 IF PLANPTS(ACTIVITY,1,3)=PINPTS(3,2)+WHALL AND PLANPTS(ACTIVITY,2,3)=
      PINPTS(3,2)+WHALL THEN GOTO 2120 ELSE GOTO 2170
2120 TRACK(M,2)=PLANPTS(ACTIVITY,1,2):TRACK(M,3)=PLANPTS(ACTIVITY,1,3)
2130 TRACK(M,4)=PLANPTS(ACTIVITY,2,2):TRACK(M,5)=PLANPTS(ACTIVITY,2,3)
2140 TRACK(M,6)=0                        :TRACK(M,7)=0
2150 TRACK(M,8)=0                        :TRACK(M,9)=0:GOTO 2220
2160 '
2170 IF PLANPTS(ACTIVITY,3,2)=PINPTS(4,1)-WHALL AND PLANPTS(ACTIVITY,2,2)=
      160-CENTER/2-WHALL THEN GOTO 2180 ELSE STOP
2180 TRACK(M,2)=0                        :TRACK(M,3)=0
2190 TRACK(M,4)=PLANPTS(ACTIVITY,2,2):TRACK(M,5)=PLANPTS(ACTIVITY,2,3)
2200 TRACK(M,6)=PLANPTS(ACTIVITY,3,2):TRACK(M,7)=PLANPTS(ACTIVITY,3,3)
2210 TRACK(M,8)=0                        :TRACK(M,9)=0:GOTO 2220
2220 '
2230 'Label floor plan
2240 '
2250 XPLACE=INT(PLANPTS(ACTIVITY,1,2)/8):YPLACE=INT(PLANPTS(ACTIVITY,1,3)/8)
2260 IF TRACK(M,3)=PINPTS(1,2) AND TRACK(M,5)=PINPTS(1,2) THEN
      LOCATE YPLACE+1,XPLACE+1:GOTO 2350 ELSE GOTO 2270
2270 IF TRACK(M,4)=PINPTS(2,1) AND TRACK(M,6)=PINPTS(2,1) THEN
      LOCATE YPLACE+1,XPLACE+2:GOTO 2350 ELSE GOTO 2280
2280 IF TRACK(M,7)=PINPTS(3,2) AND TRACK(M,9)=PINPTS(3,2) THEN
      LOCATE YPLACE+1,XPLACE+2:GOTO 2350 ELSE GOTO 2290
2290 IF TRACK(M,8)=PINPTS(4,1) AND TRACK(M,2)=PINPTS(4,1) THEN
      LOCATE YPLACE+1,XPLACE+1:GOTO 2350 ELSE GOTO 2300
2300 IF TRACK(M,9)=HALLPTS(2,2) AND TRACK(M,7)=HALLPTS(2,2) THEN
      LOCATE YPLACE-1,XPLACE+1:GOTO 2350 ELSE GOTO 2310
2310 IF TRACK(M,2)=HALLPTS(3,1) AND TRACK(M,8)=HALLPTS(3,1) THEN
      LOCATE YPLACE,XPLACE+5  :GOTO 2350 ELSE GOTO 2320
2320 IF TRACK(M,3)=HALLPTS(4,2) AND TRACK(M,5)=HALLPTS(4,2) THEN
      LOCATE YPLACE+5,XPLACE+1:GOTO 2350 ELSE GOTO 2330
2330 IF TRACK(M,4)=HALLPTS(1,1) AND TRACK(M,6)=HALLPTS(1,1) THEN
      LOCATE YPLACE,XPLACE-1  :GOTO 2350 ELSE GOTO 2340
2340 STOP
2350 IF ACTIVITY< 10 THEN PRINT USING "#"; ACTIVITY
2360 IF ACTIVITY>=10 THEN PRINT USING "##";ACTIVITY
2370 IF CFLAG=1 THEN GOTO 2490
2380 IF NPLACED<= 6 THEN GOTO 2410
2390 IF NPLACED<=12 THEN GOTO 2222
2400 IF NPLACED<=20 THEN GOTO 2470
2410 LOCATE 12+NPLACED,1:PRINT USING "#";ACTIVITY
2420 LOCATE 12+NPLACED,3:PRINT USING "\        \";TITLE$(ACTIVITY,ACTIVITY)
2430 GOTO 2490
2440 LOCATE 6+NPLACED,14:PRINT USING "#";ACTIVITY
2450 LOCATE 6+NPLACED,17:PRINT USING "\        \";TITLE$(ACTIVITY,ACTIVITY)
2460 GOTO 2490
2470 LOCATE NPLACED,28:PRINT USING "#";ACTIVITY
2480 LOCATE NPLACED,31:PRINT USING "\        \";TITLE$(ACTIVITY,ACTIVITY)
2490 CFLAG=0
2500 IF IFLAG=1 THEN GOTO 7540
2510 '
2520 IF NPLACED=N THEN GOTO 8840
2530 '
2540 'Check REL chart for heaviest relationship with placed activities
2550 'Check also for X relationships with placed activities
2560 '
```

```
2570 MAXREL=0:MINREL=-30:XFLAG=0
2580 FOR A=1 TO NPLACED
2590    FOR B=1 TO N
2600        FOR C=1 TO NPLACED
2610            IF B=TRACK(C,1) THEN GOTO 2650
2620        NEXT C
2630        IF TITLE(TRACK(A,1),B)>MAXREL THEN MAXREL=TITLE(TRACK(A,1),B):
             ACTIVITY=B:PLACED=A ELSE GOTO 2640
2640        IF TITLE(TRACK(A,1),B)=MINREL THEN XFLAG=1:TEMPACTIVITY=B:
             TEMPPLACED=A ELSE GOTO 2650
2650    NEXT B
2660 NEXT A
2670 IF XFLAG=1 THEN ACTIVITY=TEMPACTIVITY:PLACED=TEMPPLACED:GOTO 2800 ELSE
      GOTO 2680
2680 IF MAXREL>17 THEN GOTO 3040
2690 '
2700 'If no A, B or X relationships with placed activities, select largest area
2710 '
2720 MAXAREA=0
2730 FOR A=1 TO N
2740    FOR B=1 TO NPLACED
2750       IF A=TRACK(B,1) THEN GOTO 2780
2760    NEXT B
2770    IF SPACE(A,1)>MAXAREA THEN ACTIVITY=A:MAXAREA=SPACE(A,1):PLACED=1
         ELSE GOTO 2780
2780 NEXT A
2790 GOTO 3040
2800 '
2810 'Place X related activity on pinwheel in appropriate location
2820 '
2830 IF SPACE(TRACK(PLACED,1),6)<>1 THEN GOTO 2880
2840 IF SPACE(ACTIVITY,6)=4 THEN GOTO 2860
2850 IF CORNER(3)=0 THEN XDIST=HALLPTS(3,1)-PLANPTS(ACTIVITY,1,2):
                         YDIST=HALLPTS(3,2)-PLANPTS(ACTIVITY,1,3):CORNER(3)=1:
                         SPACE(ACTIVITY,6)=2:GOTO 1020 ELSE GOTO 2860
2860 IF CORNER(4)=0 THEN XDIST=HALLPTS(4,1)-PLANPTS(ACTIVITY,2,2):
                         YDIST=HALLPTS(4,2)-PLANPTS(ACTIVITY,2,3):CORNER(4)=1:
                         SPACE(ACTIVITY,6)=3:GOTO 1020 ELSE GOTO 2870
2870 IF CORNER(3)=0 THEN XDIST=HALLPTS(3,1)-PLANPTS(ACTIVITY,1,2):
                         YDIST=HALLPTS(3,2)-PLANPTS(ACTIVITY,1,3):CORNER(3)=1:
                         SPACE(ACTIVITY,6)=2:GOTO 1020 ELSE GOTO 2880
2880 IF SPACE(TRACK(PLACED,1),6)<>2 THEN GOTO 2930
2890 IF SPACE(ACTIVITY,6)=1 THEN GOTO 2910
2900 IF CORNER(4)=0 THEN XDIST=HALLPTS(4,1)-PLANPTS(ACTIVITY,2,2):
                         YDIST=HALLPTS(4,2)-PLANPTS(ACTIVITY,2,3):CORNER(4)=1:
                         SPACE(ACTIVITY,6)=3:GOTO 1020 ELSE GOTO 2910
2910 IF CORNER(1)=0 THEN XDIST=HALLPTS(1,1)-PLANPTS(ACTIVITY,3,2):
                         YDIST=HALLPTS(1,2)-PLANPTS(ACTIVITY,3,3):CORNER(1)=1:
                         SPACE(ACTIVITY,6)=4:GOTO 1020 ELSE GOTO 2920
2920 IF CORNER(4)=0 THEN XDIST=HALLPTS(4,1)-PLANPTS(ACTIVITY,2,2):
                         YDIST=HALLPTS(4,2)-PLANPTS(ACTIVITY,2,3):CORNER(4)=1:
                         SPACE(ACTIVITY,6)=3:GOTO 1020 ELSE GOTO 2930
2930 IF SPACE(TRACK(PLACED,1),6)<>3 THEN GOTO 2980
2940 IF SPACE(ACTIVITY,6)=2 THEN GOTO 2960
2950 IF CORNER(1)=0 THEN XDIST=HALLPTS(1,1)-PLANPTS(ACTIVITY,3,2):
                         YDIST=HALLPTS(1,2)-PLANPTS(ACTIVITY,3,3):CORNER(1)=1:
                         SPACE(ACTIVITY,6)=4:GOTO 1020 ELSE GOTO 2960
2960 IF CORNER(2)=0 THEN XDIST=HALLPTS(2,1)-PLANPTS(ACTIVITY,4,2):
                         YDIST=HALLPTS(2,2)-PLANPTS(ACTIVITY,4,3):CORNER(2)=1:
                         SPACE(ACTIVITY,6)=1:GOTO 1020 ELSE GOTO 2920
```

```
2970 IF CORNER(1)=0 THEN XDIST=HALLPTS(1,1)-PLANPTS(ACTIVITY,3,2):
                         YDIST=HALLPTS(1,2)-PLANPTS(ACTIVITY,3,3):CORNER(1)=1:
                         SPACE(ACTIVITY,6)=4:GOTO 1020 ELSE GOTO 2930
2980 IF SPACE(TRACK(PLACED,1),6)<>4 THEN 3030
2990 IF SPACE(ACTIVITY,6)=3 THEN GOTO 3010
3000 IF CORNER(2)=0 THEN XDIST=HALLPTS(2,1)-PLANPTS(ACTIVITY,4,2):
                         YDIST=HALLPTS(2,2)-PLANPTS(ACTIVITY,4,3):CORNER(2)=1:
                         SPACE(ACTIVITY,6)=1:GOTO 1020 ELSE GOTO 3010
3010 IF CORNER(3)=0 THEN XDIST=HALLPTS(3,1)-PLANPTS(ACTIVITY,1,2):
                         YDIST=HALLPTS(3,2)-PLANPTS(ACTIVITY,1,3):CORNER(3)=1:
                         SPACE(ACTIVITY,6)=2:GOTO 1020 ELSE GOTO 3020
3020 IF CORNER(2)=0 THEN XDIST=HALLPTS(2,1)-PLANPTS(ACTIVITY,4,2):
                         YDIST=HALLPTS(2,2)-PLANPTS(ACTIVITY,4,3):CORNER(2)=1:
                         SPACE(ACTIVITY,6)=1:GOTO 1020 ELSE GOTO 3030
3030 STOP
3040 '
3050 'Place activity on pinwheel next to appropriate activity
3060 '
3070 NLENGTH=ABS(PLANPTS(ACTIVITY,4,2)-PLANPTS(ACTIVITY,3,2))
3080 ELENGTH=ABS(PLANPTS(ACTIVITY,1,3)-PLANPTS(ACTIVITY,4,3))
3090 SLENGTH=ABS(PLANPTS(ACTIVITY,1,2)-PLANPTS(ACTIVITY,2,2))
3100 WLENGTH=ABS(PLANPTS(ACTIVITY,2,3)-PLANPTS(ACTIVITY,3,3))
3110 '
3120 IF (SPACE(ACTIVITY,6)=0 OR SPACE(ACTIVITY,6)=1) AND
      SPACE(TRACK(PLACED,1),6)=1 THEN SFLAG=1:SPACE(ACTIVITY,6)=1:GOTO 3300
      ELSE GOTO 3130
3130 IF (SPACE(ACTIVITY,6)=0 OR SPACE(ACTIVITY,6)=3) AND
      SPACE(TRACK(PLACED,1),6)=3 THEN SFLAG=2:SPACE(ACTIVITY,6)=3:GOTO 4240
      ELSE GOTO 3140
3140 IF SPACE(ACTIVITY,6)=3 AND SPACE(TRACK(PLACED,1),6)=1 THEN
      SFLAG=3:GOTO 4920 ELSE GOTO 3150
3150 IF SPACE(ACTIVITY,6)=1 AND SPACE(TRACK(PLACED,1),6)=3 THEN
      SFLAG=4:GOTO 4000 ELSE GOTO 3160
3160 IF (SPACE(ACTIVITY,6)=0 OR SPACE(ACTIVITY,6)=2) AND
      SPACE(TRACK(PLACED,1),6)=2 THEN SFLAG=5:SPACE(ACTIVITY,6)=2:GOTO 5170
      ELSE GOTO 3170
3170 IF (SPACE(ACTIVITY,6)=0 OR SPACE(ACTIVITY,6)=4) AND
      SPACE(TRACK(PLACED,1),6)=4 THEN SFLAG=6:SPACE(ACTIVITY,6)=4:GOTO 6090
      ELSE GOTO 3180
3180 IF SPACE(ACTIVITY,6)=4 AND SPACE(TRACK(PLACED,1),6)=2 THEN
      SFLAG=7:GOTO 6780 ELSE GOTO 3190
3190 IF SPACE(ACTIVITY,6)=2 AND SPACE(TRACK(PLACED,1),6)=4 THEN
      SFLAG=8:GOTO 5850 ELSE GOTO 3200
3200 IF SPACE(ACTIVITY,6)=1 AND SPACE(TRACK(PLACED,1),6)=4 THEN
     SFLAG=9:GOTO 3300 ELSE GOTO 3210
3210 IF SPACE(ACTIVITY,6)=1 AND SPACE(TRACK(PLACED,1),6)=2 THEN
      SFLAG=10:GOTO 3300 ELSE GOTO 3220
3220 IF SPACE(ACTIVITY,6)=3 AND SPACE(TRACK(PLACED,1),6)=4 THEN
      SFLAG=11:GOTO 4240 ELSE GOTO 3230
3230 IF SPACE(ACTIVITY,6)=3 AND SPACE(TRACK(PLACED,1),6)=2 THEN
      SFLAG=12:GOTO 4240 ELSE GOTO 3240
3240 IF SPACE(ACTIVITY,6)=4 AND SPACE(TRACK(PLACED,1),6)=3 THEN
      SFLAG=13:GOTO 6090 ELSE GOTO 3250
3250 IF SPACE(ACTIVITY,6)=4 AND SPACE(TRACK(PLACED,1),6)=1 THEN
      SFLAG=14:GOTO 6090 ELSE GOTO 3260
3260 IF SPACE(ACTIVITY,6)=2 AND SPACE(TRACK(PLACED,1),6)=1 THEN
      SFLAG=15:GOTO 5170 ELSE GOTO 3270
3270 IF SPACE(ACTIVITY,6)=2 AND SPACE(TRACK(PLACED,1),6)=3 THEN
      SFLAG=16:GOTO 5170 ELSE GOTO 3280
3280 STOP
```

```
3290 '
3300 'Pinwheel placement
3310 '
3320 'North side check - courtyard
3330 '
3340 FOR A=1 TO N
3350    IF TRACK(A,2)=PINPTS(1,1) AND TRACK(A,3)=PINPTS(1,2) THEN GOTO 3420
3360    IF PLANPTS(A,1,2)<=PINPTS(1,1)+NLENGTH AND PLANPTS(A,1,2)>=PINPTS(1,1)
         AND PLANPTS(A,1,3)<=PINPTS(1,2)+WLENGTH AND PLANPTS(A,1,3)>=PINPTS(1,2)
         THEN GOTO 3580
3370 NEXT A
3380 IF PINPTS(1,1)+NLENGTH>PINPTS(2,1) THEN GOTO 3420
3390 XDIST=PINPTS(1,1)-PLANPTS(ACTIVITY,1,2)
3400 YDIST=PINPTS(1,2)-PLANPTS(ACTIVITY,1,3):GOTO 1020
3410 '
3420 '
3430 AFLAG=0:OVERFLAG=0:ENDFLAG=0
3440    IF AFLAG=0 THEN GOTO 3470
3450    IF OVERFLAG=0 THEN OVERFLAG=1:PLACED=1 ELSE GOTO 3460
3460    IF PLACED>NPLACED THEN ENDFLAG=1:GOTO 3560 ELSE GOTO 3470
3470    IF TRACK(PLACED,3)<>PINPTS(1,2) THEN GOTO 3560
3480    IF TRACK(PLACED,4)+NLENGTH>PINPTS(2,1) THEN GOTO 3560
3490    FOR A=1 TO NPLACED
3500      IF TRACK(A,3)<>PINPTS(1,2) THEN GOTO 3530
3510      IF TRACK(A,2)>=TRACK(PLACED,4) AND TRACK(A,2)<=TRACK(PLACED,4)+NLENGTH
           THEN GOTO 3560
3520      IF TRACK(A,2)=TRACK(PLACED,4) AND TRACK(A,4)-TRACK(A,2)>NLENGTH
           THEN GOTO 3560
3530    NEXT A
3540    XDIST=TRACK(PLACED,4)-PLANPTS(ACTIVITY,1,2)
3550    YDIST=TRACK(PLACED,5)-PLANPTS(ACTIVITY,1,3):GOTO 1020
3560 IF ENDFLAG=1 THEN GOTO 3580 ELSE PLACED=PLACED+1
3570 GOTO 3450
3580 '
3590 'North side check
3600 '
3610 FOR A=1 TO NPLACED
3620   IF TRACK(A,8)=PINPTS(1,1) AND TRACK(A,9)=PINPTS(1,2)-WHALL THEN GOTO 3670

3630 NEXT A
3640 IF PINPTS(1,1)+NLENGTH>HALLPTS(2,1) THEN GOTO 3800
3650 XDIST=PINPTS(1,1)-PLANPTS(ACTIVITY,4,2)
3660 YDIST=PINPTS(1,2)-WHALL-PLANPTS(ACTIVITY,4,3):GOTO 1020
3670 BFLAG=0:OVERFLAG=0:ENDFLAG=0
3680    IF BFLAG=0 THEN GOTO 3710
3690    IF OVERFLAG=0 THEN OVERFLAG=1:PLACED=1 ELSE GOTO 3700
3700    IF PLACED>NPLACED THEN ENDFLAG=1:GOTO 3800 ELSE GOTO 3710
3710    IF TRACK(PLACED,6)+NLENGTH>HALLPTS(2,1) THEN GOTO 3800
3720    IF TRACK(PLACED,7)<>HALLPTS(2,2) THEN GOTO 3800
3730    FOR A=1 TO NPLACED
3740      IF TRACK(A,7)<>HALLPTS(2,2) THEN GOTO 3770
3750      IF TRACK(A,8)>=TRACK(PLACED,6) AND TRACK(A,8)<=TRACK(PLACED,6)+NLENGTH
           THEN GOTO 3980
3760      IF TRACK(A,8)=TRACK(PLACED,6) AND TRACK(A,6)-TRACK(A,8)>NLENGTH
           THEN GOTO 3800
3770    NEXT A
3780    XDIST=TRACK(PLACED,6)-PLANPTS(ACTIVITY,4,2)
3790    YDIST=TRACK(PLACED,7)-PLANPTS(ACTIVITY,4,3):GOTO 1020
3800 IF ENDFLAG=1 THEN GOTO 3820 ELSE PLACED=PLACED+1
3810 GOTO 3690
```

```
3820 '
3830 'North side check - upper hall
3840 '
3850 DFLAG=0:OVERFLAG=0:ENDFLAG=0
3860    IF DFLAG=0 THEN GOTO 3890
3870    IF OVERFLAG=0 THEN OVERFLAG=1:PLACED=1 ELSE GOTO 3880
3880    IF PLACED>NPLACED THEN ENDFLAG=1:GOTO 3980 ELSE GOTO 3890
3890    IF TRACK(PLACED,2)<>PINPTS(1,1) THEN GOTO 3980
3900    IF TRACK(PLACED,3)-WLENGTH<HALLPTS(1,2) THEN GOTO 3980
3910    FOR A=1 TO NPLACED
3920      IF TRACK(A,2)<>PINPTS(1,1) THEN GOTO 3950
3930      IF TRACK(A,9)<=TRACK(PLACED,3) AND TRACK(A,9)>=TRACK(PLACED,3)-ELENGTH
           THEN GOTO 3980
3940      IF TRACK(A,9)=TRACK(PLACED,3) AND TRACK(A,9)-TRACK(A,3)>ELENGTH
           THEN GOTO 3980
3950    NEXT A
3960    XDIST=TRACK(PLACED,2)-PLANPTS(ACTIVITY,4,2)
3970    YDIST=TRACK(PLACED,3)-PLANPTS(ACTIVITY,4,3):GOTO 1020
3980 IF ENDFLAG=1 THEN GOTO 4000 ELSE PLACED=PLACED+1
3990 GOTO 3870
4000 '
4010 'North side check - separate hall
4020 '
4030 FOR A=1 TO NPLACED
4040    IF TRACK(A,6)=HALLPTS(1,1) AND TRACK(A,7)=PINPTS(4,2) THEN GOTO 4080
4050 NEXT A
4060 XDIST=HALLPTS(1,1)-PLANPTS(ACTIVITY,3,2)
4070 YDIST=PINPTS(4,2)-PLANPTS(ACTIVITY,3,3):GOTO 1020
4080 EFLAG=0:OVERFLAG=0:ENDFLAG=0
4090    IF EFLAG=0 THEN GOTO 4120
4100    IF OVERFLAG=0 THEN OVERFLAG=1:PLACED=1 ELSE GOTO 4110
4110    IF PLACED>NPLACED THEN ENDFLAG=1:GOTO 4210 ELSE GOTO 4120
4120    IF TRACK(PLACED,9)<>PINPTS(4,2) THEN GOTO 4210
4130    IF TRACK(PLACED,8)-SLENGTH<HALLPTS(4,1) THEN GOTO 4210
4140    FOR A=1 TO NPLACED
4150      IF TRACK(A,9)<>PINPTS(4,2) THEN GOTO 4180
4160      IF TRACK(A,6)<=TRACK(PLACED,8) AND TRACK(A,6)>=TRACK(PLACED,8)-SLENGTH
           THEN GOTO 4210
4170      IF TRACK(A,6)=TRACK(PLACED,8) AND TRACK(A,6)-TRACK(A,8)>SLENGTH
           THEN GOTO 4210
4180    NEXT A
4190    XDIST=TRACK(PLACED,8)-PLANPTS(ACTIVITY,3,2)
4200    YDIST=TRACK(PLACED,9)-PLANPTS(ACTIVITY,3,3):GOTO 1020
4210 IF ENDFLAG=1 THEN GOTO 4230 ELSE PLACED=PLACED+1
4220 GOTO 4100
4230 IF SFLAG=4 THEN GOTO 3310 ELSE GOTO 7020
4240 '
4250 'South side check - courtyard
4260 '
4270 FOR A=1 TO N
4280    IF TRACK(A,6)=PINPTS(3,1) AND TRACK(A,7)=PINPTS(3,2) THEN GOTO 4340
4290     IF PLANPTS(A,3,2)>=PINPTS(3,1)-SLENGTH AND PLANPTS(A,3,2)<=PINPTS(3,1)
          AND PLANPTS(A,3,3)>=PINPTS(3,2)-ELENGTH AND PLANPTS(A,3,3)<=PINPTS(3,2)
          THEN GOTO 4500
4300 NEXT A
4310 IF PINPTS(3,1)-SLENGTH<PINPTS(4,1) THEN GOTO 4340
4320 XDIST=PINPTS(3,1)-PLANPTS(ACTIVITY,3,2)
4330 YDIST=PINPTS(3,2)-PLANPTS(ACTIVITY,3,3):GOTO 1020
4340 '
4350 FFLAG=0:OVERFLAG=0:ENDFLAG=0
```

```
4360    IF FFLAG=0 THEN GOTO 4390
4370    IF OVERFLAG=0 THEN OVERFLAG=1:PLACED=1 ELSE GOTO 4380
4380    IF PLACED>NPLACED THEN ENDFLAG=1:GOTO 4480 ELSE GOTO 4390
4390    IF TRACK(PLACED,9)<>PINPTS(3,2) THEN GOTO 4480
4400    IF TRACK(PLACED,8)-SLENGTH<PINPTS(4,1) THEN GOTO 4480
4410    FOR A=1 TO NPLACED
4420      IF TRACK(A,9)<>PINPTS(3,2) THEN GOTO 4450
4430      IF TRACK(A,6)<=TRACK(PLACED,8) AND TRACK(A,6)>=TRACK(PLACED,8)-SLENGTH
           THEN GOTO 4480
4440      IF TRACK(A,6)=TRACK(A,8) AND TRACK(A,6)-TRACK(A,8)>SLENGTH
           THEN GOTO 4480
4450    NEXT A
4460    XDIST=TRACK(PLACED,8)-PLANPTS(ACTIVITY,3,2)
4470    YDIST=TRACK(PLACED,9)-PLANPTS(ACTIVITY,3,3):GOTO 1020
4480 IF ENDFLAG=1 THEN GOTO 4500 ELSE PLACED=PLACED+1
4490 GOTO 4370
4500 '
4510 'South side check
4520 '
4530 FOR A=1 TO NPLACED
4540    IF TRACK(A,4)=PINPTS(3,1) AND TRACK(A,5)=HALLPTS(4,2) THEN GOTO 4590
4550 NEXT A
4560 XDIST=PINPTS(3,1)-PLANPTS(ACTIVITY,2,2)
4570 YDIST=PINPTS(3,2)+WHALL-PLANPTS(ACTIVITY,2,3):GOTO 1020
4580 '
4590 GFLAG=0:OVERFLAG=0:ENDFLAG=0
4600    IF GFLAG=0 THEN GOTO 4630
4610    IF OVERFLAG=0 THEN OVERFLAG=1:PLACED=1 ELSE GOTO 4620
4620    IF PLACED>NPLACED THEN ENDFLAG=1:GOTO 4720 ELSE GOTO 4630
4630    IF TRACK(PLACED,3)<>HALLPTS(4,2) THEN GOTO 4720
4640    IF TRACK(PLACED,2)-SLENGTH<HALLPTS(4,1) THEN GOTO 4720
4650    FOR A=1 TO NPLACED
4660      IF TRACK(A,3)<>HALLPTS(4,2) THEN GOTO 4690
4670      IF TRACK(A,4)<=TRACK(PLACED,2) AND TRACK(A,4)>=TRACK(PLACED,2)-SLENGTH
           THEN GOTO 4720
4680      IF TRACK(A,4)=TRACK(PLACED,2) AND TRACK(A,4)-TRACK(A,2)>SLENGTH
           THEN GOTO 4720
4690    NEXT A
4700    XDIST=TRACK(PLACED,2)-PLANPTS(ACTIVITY,2,2)
4710    YDIST=TRACK(PLACED,3)-PLANPTS(ACTIVITY,2,3):GOTO 1020
4720 IF ENDFLAG=1 THEN GOTO 4740 ELSE PLACED=PLACED+1
4730 GOTO 4610
4740 '
4750 'South side check - lower wall
4760 '
4770 HFLAG=0:OVERFLAG=0:ENDFLAG=0
4780    IF HFLAG=0 THEN GOTO 4810
4790    IF OVERFLAG=0 THEN OVERFLAG=1:PLACED=1 ELSE GOTO 4800
4800    IF PLACED>NPLACED THEN ENDFLAG=1:GOTO 4900 ELSE GOTO 4810
4810    IF TRACK(PLACED,4)<>PINPTS(3,1) THEN GOTO 4900
4820    IF TRACK(PLACED,7)+ELENGTH>HALLPTS(3,2) THEN GOTO 4910
4830    FOR A=1 TO NPLACED
4840      IF TRACK(A,4)<>PINPTS(3,1) THEN GOTO 4870
4850      IF TRACK(A,5)>=TRACK(PLACED,7) AND TRACK(A,5)<=TRACK(PLACED,7)+ELENGTH
           THEN GOTO 4900
4860      IF TRACK(A,5)=TRACK(PLACED,7) AND TRACK(A,7)-TRACK(A,5)>ELENGTH
           THEN GOTO 4900
4870    NEXT A
4880    XDIST=TRACK(PLACED,7)-PLANPTS(ACTIVITY,2,2)
4890    YDIST=TRACK(PLACED,7)-PLANPTS(ACTIVITY,2,3):GOTO 1020
```

```
4900 IF ENDFLAG=1 THEN GOTO 4920 ELSE PLACED=PLACED+1
4910 GOTO 4790
4920 '
4930 'South side check - separate hall
4940 '
4950 FOR A=1 TO NPLACED
4960   IF TRACK(A,2)=PINPTS(2,1)+WHALL AND TRACK(A,3)=PINPTS(2,2) THEN GOTO 5000
4970 NEXT A
4980 XDIST=PINPTS(2,1)+WHALL-PLANPTS(ACTIVITY,1,2)
4990 YDIST=PINPTS(2,2)-PLANPTS(ACTIVITY,1,3):GOTO 1020
5000 '
5010 GFLAG=0:OVERFLAG=0:ENDFLAG=0
5020    IF GFLAG=0 THEN GOTO 5050
5030    IF OVERFLAG=0 THEN OVERFLAG=1:PLACED=1 ELSE GOTO 5040
5040    IF PLACED>NPLACED THEN ENDFLAG=1:GOTO 5140 ELSE GOTO 5050
5050    IF TRACK(PLACED,3)<>PINPTS(2,2) THEN GOTO 5140
5060    IF TRACK(PLACED,4)+NLENGTH>HALLPTS(2,1) THEN GOTO 5140
5070    FOR A=1 TO NPLACED
5080      IF TRACK(A,3)<>PINPTS(2,2) THEN GOTO 5110
5090      IF TRACK(A,2)>=TRACK(PLACED,4) AND TRACK(A,2)<=TRACK(PLACED,4)+NLENGTH
           THEN GOTO 5140
5100      IF TRACK(A,2)=TRACK(PLACED,4) AND TRACK(A,4)-TRACK(A,2)>NLENGTH
           THEN GOTO 5140
5110    NEXT A
5120    XDIST=TRACK(PLACED,4)-PLANPTS(ACTIVITY,1,2)
5130    YDIST=TRACK(PLACED,5)-PLANPTS(ACTIVITY,1,3):GOTO 1020
5140 IF ENDFLAG=1 THEN GOTO 5160 ELSE PLACED=PLACED+1
5150 GOTO 5030
5160 IF SFLAG=3 THEN GOTO 4240 ELSE GOTO 7020
5170 '
5180 'East side check - courtyard
5190 '
5200 FOR A=1 TO N
5210    IF TRACK(A,4)=PINPTS(2,1) AND TRACK(A,5)=PINPTS(2,2) THEN GOTO 5280
5220    IF PLANPTS(A,2,2)>=PINPTS(2,1)-NLENGTH AND PLANPTS(A,2,2)<=PINPTS(2,1)
         AND PLANPTS(A,2,3)<=PINPTS(2,2)+ELENGTH AND PLANPTS(A,2,3)>=PINPTS(2,2)
         THEN GOTO 5450
5230 NEXT A
5240 IF PINPTS(2,1)-NLENGTH<PINPTS(1,1) THEN GOTO 5280
5250 XDIST=PINPTS(2,1)-PLANPTS(ACTIVITY,2,2)
5260 YDIST=PINPTS(2,2)-PLANPTS(ACTIVITY,2,3):GOTO 1020
5270 '
5280 HFLAG=0:OVERFLAG=0:ENDFLAG=0
5290    IF HFLAG=0 THEN GOTO 5320
5300    IF OVERFLAG=0 THEN OVERFLAG=1:PLACED=1 ELSE GOTO 5310
5310    IF PLACED>NPLACED THEN ENDFLAG=1:GOTO 5410 ELSE GOTO 5320
5320    IF TRACK(PLACED,4)<>PINPTS(2,1) THEN GOTO 5410
5330    IF TRACK(PLACED,7)+ELENGTH>PINPTS(3,2) THEN GOTO 5410
5340    FOR A=1 TO NPLACED
5350      IF TRACK(A,4)<>PINPTS(2,1) THEN GOTO 5380
5360      IF TRACK(A,5)>=TRACK(PLACED,7) AND TRACK(A,5)<=TRACK(PLACED,7)+ELENGTH
           THEN GOTO 5410
5370      IF TRACK(A,5)=TRACK(PLACED,7) AND TRACK(A,7)-TRACK(A,5)>ELENGTH
           THEN GOTO 5410
5380    NEXT A
5390    XDIST=TRACK(PLACED,6)-PLANPTS(ACTIVITY,2,2)
5400    YDIST=TRACK(PLACED,7)-PLANPTS(ACTIVITY,2,3):GOTO 1020
5410 IF ENDFLAG=1 THEN GOTO 5430 ELSE PLACED=PLACED+1
5420 GOTO 5300
```

```
5430 '
5440 'East side check
5450 '
5460 FOR A=1 TO NPLACED
5470   IF TRACK(A,2)=PINPTS(2,1)+WHALL AND TRACK(A,3)=PINPTS(2,2) THEN GOTO 5520
5480 NEXT A
5490 IF PINPTS(2,2)+ELENGTH>HALLPTS(3,2) THEN GOTO 5650
5500 XDIST=PINPTS(2,1)+WHALL-PLANPTS(ACTIVITY,1,2)
5510 YDIST=PINPTS(2,2)-PLANPTS(ACTIVITY,1,3):GOTO 1020
5520 JFLAG=0:OVERFLAG=0:ENDFLAG=0
5530    IF JFLAG=0 THEN GOTO 5560
5540    IF OVERFLAG=0 THEN OVERFLAG=1:PLACED=1 ELSE GOTO 5550
5550    IF PLACED>NPLACED THEN ENDFLAG=1:GOTO 5650 ELSE GOTO 5560
5560    IF TRACK(PLACED,9)+ELENGTH>HALLPTS(3,2) THEN GOTO 5650
5570    IF TRACK(PLACED,8)<>HALLPTS(3,1) THEN GOTO 5650
5580    FOR A=1 TO NPLACED
5590      IF TRACK(A,8)<>HALLPTS(3,1) THEN GOTO 5620
5600      IF TRACK(A,3)>=TRACK(PLACED,9) AND TRACK(A,3)<=TRACK(PLACED,9)+ELENGTH
           THEN GOTO 5830
5610      IF TRACK(A,3)=TRACK(PLACED,9) AND TRACK(A,9)-TRACK(A,3)>NLENGTH
           THEN GOTO 5650
5620    NEXT A
5630    XDIST=TRACK(PLACED,8)-PLANPTS(ACTIVITY,1,2)
5640    YDIST=TRACK(PLACED,9)-PLANPTS(ACTIVITY,1,3):GOTO 1020
5650 IF ENDFLAG=1 THEN GOTO 5670 ELSE PLACED=PLACED+1
5660 GOTO 5540
5670 '
5680 'East side check - upper hall
5690 '
5700 KFLAG=0:OVERFLAG=0:ENDFLAG=0
5710    IF KFLAG=0 THEN GOTO 5740
5720    IF OVERFLAG=0 THEN OVERFLAG=1:PLACED=1 ELSE GOTO 5730
5730    IF PLACED>NPLACED THEN ENDFLAG=1:GOTO 5830 ELSE GOTO 5740
5740    IF TRACK(PLACED,5)<>PINPTS(2,2) THEN GOTO 5830
5750    IF TRACK(PLACED,4)+NLENGTH>HALLPTS(2,1) THEN GOTO 5840
5760    FOR A=1 TO NPLACED
5770      IF TRACK(A,5)<>PINPTS(2,2) THEN GOTO 5800
5780      IF TRACK(A,2)>=TRACK(PLACED,4) AND TRACK(A,2)<=TRACK(PLACED,4)+NLENGTH
           THEN GOTO 5830
5790      IF TRACK(A,2)=TRACK(PLACED,4) AND TRACK(A,4)-TRACK(A,2)>ELENGTH
           THEN GOTO 5830
5800    NEXT A
5810    XDIST=TRACK(PLACED,4)-PLANPTS(ACTIVITY,1,2)
5820    YDIST=TRACK(PLACED,5)-PLANPTS(ACTIVITY,1,3):GOTO 1020
5830 IF ENDFLAG=1 THEN GOTO 5850 ELSE PLACED=PLACED+1
5840 GOTO 5720
5850 '
5860 'East side check - separate hall
5870 '
5880 FOR A=1 TO NPLACED
5890    IF TRACK(A,8)=PINPTS(1,1) AND TRACK(A,9)=HALLPTS(2,2) THEN GOTO 5930
5900 NEXT A
5910 XDIST=PINPTS(1,1)-PLANPTS(ACTIVITY,4,2)
5920 YDIST=HALLPTS(2,2)-PLANPTS(ACTIVITY,4,3):GOTO 1020
5930 LFLAG=0:OVERFLAG=0:ENDFLAG=0
5940    IF LFLAG=0 THEN GOTO 5970
5950    IF OVERFLAG=0 THEN OVERFLAG=1:PLACED=1 ELSE GOTO 5960
5960    IF PLACED>NPLACED THEN ENDFLAG=1:GOTO 6060 ELSE GOTO 5970
5970    IF TRACK(PLACED,2)<>PINPTS(2,1) THEN GOTO 6060
```

```
5980     IF TRACK(PLACED,3)-WLENGTH<HALLPTS(1,2) THEN GOTO 6060
5990     FOR A=1 TO NPLACED
6000       IF TRACK(A,2)<>PINPTS(2,1) THEN GOTO 6030
6010       IF TRACK(A,9)<=TRACK(PLACED,3) AND TRACK(A,9)>=TRACK(PLACED,3)-ELENGTH
            THEN GOTO 6060
6020       IF TRACK(A,9)=TRACK(PLACED,3) AND TRACK(A,9)-TRACK(A,3)>ELENGTH
            THEN GOTO 6060
6030     NEXT A
6040     XDIST=TRACK(PLACED,2)-PLANPTS(ACTIVITY,4,2)
6050     YDIST=TRACK(PLACED,3)-PLANPTS(ACTIVITY,4,3):GOTO 1020
6060 IF ENDFLAG=1 THEN GOTO 6080 ELSE PLACED=PLACED+1
6070 GOTO 5950
6080 IF SFLAG=7 THEN GOTO 5170 ELSE GOTO 7020
6090 '
6100 'West side check - courtyard
6110 '
6120 FOR A=1 TO N
6130     IF TRACK(A,8)=PINPTS(4,1) AND TRACK(A,9)=PINPTS(4,2) THEN GOTO 6200
6140     IF PLANPTS(A,4,2)<=PINPTS(4,1)+SLENGTH AND PLANPTS(A,4,2)>=PINPTS(4,1)
          AND PLANPTS(A,4,3)>=PINPTS(4,2)-WLENGTH AND PLANPTS(A,4,3)<=PINPTS(4,2)
          THEN GOTO 6360
6150 NEXT A
6160 IF PINPTS(4,2)-WLENGTH<PINPTS(1,2) THEN GOTO 6200
6170 XDIST=PINPTS(4,1)-PLANPTS(ACTIVITY,4,2)
6180 YDIST=PINPTS(4,2)-PLANPTS(ACTIVITY,4,3):GOTO 1020
6190 '
6200 '
6210 MFLAG=0:OVERFLAG=0:ENDFLAG=0
6220     IF MFLAG=0 THEN GOTO 6250
6230     IF OVERFLAG=0 THEN OVERFLAG=1:PLACED=1 ELSE GOTO 6240
6240     IF PLACED>NPLACED THEN ENDFLAG=1:GOTO 6340 ELSE GOTO 6250
6250     IF TRACK(PLACED,8)<>PINPTS(4,1) THEN GOTO 6340
6260     IF TRACK(PLACED,3)-WLENGTH<PINPTS(1,2) THEN GOTO 6340
6270     FOR A=1 TO NPLACED
6280       IF TRACK(A,8)<>PINPTS(4,1) THEN GOTO 6310
6290       IF TRACK(A,9)<=TRACK(PLACED,3) AND TRACK(A,9)>=TRACK(PLACED,3)-WLENGTH
            THEN GOTO 6340
6300       IF TRACK(A,9)=TRACK(PLACED,3) AND TRACK(A,9)-TRACK(A,3)>WLENGTH
            THEN GOTO 6340
6310     NEXT A
6320     XDIST=TRACK(PLACED,2)-PLANPTS(ACTIVITY,4,2)
6330     YDIST=TRACK(PLACED,3)-PLANPTS(ACTIVITY,4,3):GOTO 1020
6340 IF ENDFLAG=1 THEN GOTO 6360 ELSE PLACED=PLACED+1
6350 GOTO 6230
6360 '
6370 'West side check
6380 '
6390 FOR A=1 TO NPLACED
6400   IF TRACK(A,6)=HALLPTS(1,1) AND TRACK(A,7)=PINPTS(4,2) THEN GOTO 6450
6410 NEXT A
6420 IF PINPTS(4,2)-WLENGTH<HALLPTS(1,2) THEN GOTO 6580
6430 XDIST=HALLPTS(1,1)-PLANPTS(ACTIVITY,3,2)
6440 YDIST=PINPTS(4,2)-PLANPTS(ACTIVITY,3,3):GOTO 1020
6450 NFLAG=0:OVERFLAG=0:ENDFLAG=0
6460     IF NFLAG=0 THEN GOTO 6490
6470     IF OVERFLAG=0 THEN OVERFLAG=1:PLACED=1 ELSE GOTO 6480
6480     IF PLACED>NPLACED THEN ENDFLAG=1:GOTO 6580 ELSE GOTO 6490
6490     IF TRACK(PLACED,5)-WLENGTH<HALLPTS(1,2) THEN GOTO 6580
6500     IF TRACK(PLACED,4)<>HALLPTS(1,1) THEN GOTO 6580
6510     FOR A=1 TO NPLACED
```

```
6520       IF TRACK(A,4)<>HALLPTS(1,1) THEN GOTO 6550
6530       IF TRACK(A,7)<=TRACK(PLACED,5) AND TRACK(A,7)>=TRACK(PLACED,5)-WLENGTH
            THEN GOTO 6580
6540       IF TRACK(A,7)=TRACK(PLACED,5) AND TRACK(A,7)-TRACK(A,5)>WLENGTH
            THEN GOTO 6580
6550     NEXT A
6560     XDIST=TRACK(PLACED,4)-PLANPTS(ACTIVITY,3,2)
6570     YDIST=TRACK(PLACED,5)-PLANPTS(ACTIVITY,3,3):GOTO 1020
6580 IF ENDFLAG=1 THEN GOTO 6600 ELSE PLACED=PLACED+1
6590 GOTO 6470
6600 '
6610 'West side check - upper hall
6620 '
6630 OFLAG=0:OVERFLAG=0:ENDFLAG=0
6640     IF OFLAG=0 THEN GOTO 6670
6650     IF OVERFLAG=0 THEN OVERFLAG=1:PLACED=1 ELSE GOTO 6660
6660     IF PLACED>NPLACED THEN ENDFLAG=1:GOTO 6760 ELSE GOTO 6670
6670     IF TRACK(PLACED,9)<>PINPTS(4,2) THEN GOTO 6760
6680     IF TRACK(PLACED,8)-SLENGTH<HALLPTS(4,1) THEN GOTO 6760
6690     FOR A=1 TO NPLACED
6700       IF TRACK(A,9)<>PINPTS(4,2) THEN GOTO 6730
6710       IF TRACK(A,6)<=TRACK(PLACED,8) AND TRACK(A,6)>=TRACK(PLACED,8)-SLENGTH
            THEN GOTO 6760
6720       IF TRACK(A,6)=TRACK(PLACED,8) AND TRACK(A,6)-TRACK(A,8)>SLENGTH
            THEN GOTO 6760
6730     NEXT A
6740     XDIST=TRACK(PLACED,8)-PLANPTS(ACTIVITY,3,2)
6750     YDIST=TRACK(PLACED,9)-PLANPTS(ACTIVITY,3,3):GOTO 1020
6760 IF ENDFLAG=1 THEN GOTO 6780 ELSE PLACED=PLACED+1
6770 GOTO 6650
6780 '
6790 'West side check - separate hall
6800 '
6810 FOR A=1 TO NPLACED
6820     IF TRACK(A,4)=PINPTS(3,1) AND TRACK(A,5)=HALLPTS(4,2) THEN GOTO 6860
6830 NEXT A
6840 XDIST=PINPTS(3,1)-PLANPTS(ACTIVITY,2,2)
6850 YDIST=HALLPTS(4,2)-PLANPTS(ACTIVITY,2,3):GOTO 1020
6860 PFLAG=0:OVERFLAG=0:ENDFLAG=0
6870     IF PFLAG=0 THEN GOTO 6900
6880     IF OVERFLAG=0 THEN OVERFLAG=1:PLACED=1 ELSE GOTO 6890
6890     IF PLACED>NPLACED THEN ENDFLAG=1:GOTO 6990 ELSE GOTO 6900
6900     IF TRACK(PLACED,6)<>PINPTS(3,1) THEN GOTO 6990
6910     IF TRACK(PLACED,7)+ELENGTH>HALLPTS(3,2) THEN GOTO 6990
6920     FOR A=1 TO NPLACED
6930       IF TRACK(A,6)<>PINPTS(3,1) THEN GOTO 6960
6940       IF TRACK(A,5)>=TRACK(PLACED,7) AND TRACK(A,5)<=TRACK(PLACED,7)+ELENGTH
            THEN GOTO 6990
6950       IF TRACK(A,5)=TRACK(PLACED,7) AND TRACK(A,7)-TRACK(A,5)>ELENGTH
            THEN GOTO 6990
6960     NEXT A
6970     XDIST=TRACK(PLACED,6)-PLANPTS(ACTIVITY,2,2)
6980     YDIST=TRACK(PLACED,7)-PLANPTS(ACTIVITY,2,3):GOTO 1020
6990 IF ENDFLAG=1 THEN GOTO 7010 ELSE PLACED=PLACED+1
7000 GOTO 6880
7010 IF SFLAG=8 THEN GOTO 6090 ELSE GOTO 7020
7020 '
7030 'No locations available on desired radial side
7040 '
7050 XDIST=250-PLANPTS(ACTIVITY,1,2)
```

```
7060 YDIST=150-PLANPTS(ACTIVITY,1,3)
7070 FOR A=1 TO SPACE(ACTIVITY,11)
7080    PLANPTS(ACTIVITY,A,2)=PLANPTS(ACTIVITY,A,2)+XDIST
7090    PLANPTS(ACTIVITY,A,3)=PLANPTS(ACTIVITY,A,3)+YDIST
7100 NEXT A
7110 FLAG=1:I=1:COUNT=0
7120 WHILE FLAG
7130     I=PLANPTS(ACTIVITY,I,1)
7140     LINE (PLANPTS(ACTIVITY,PLANPTS(ACTIVITY,I,1),2),PLANPTS(ACTIVITY,
           PLANPTS(ACTIVITY,I,1),3))-(PLANPTS(ACTIVITY,PLANPTS(ACTIVITY,I,4),2)
           ,PLANPTS(ACTIVITY,PLANPTS(ACTIVITY,I,4),3)),1
7150     I=PLANPTS(ACTIVITY,I,4)
7160     COUNT=COUNT+1
7170     IF COUNT>=SPACE(ACTIVITY,11) THEN FLAG=0
7180 WEND
7190 LOCATE 21,1:PRINT USING "#";ACTIVITY
7200 LOCATE 21,4:PRINT TITLE$(ACTIVITY,ACTIVITY)
7210 LOCATE 22,1:PRINT TITLE$(ACTIVITY,PLACED)
7220 LOCATE 22,2:PRINT "-RELATIONSHIP WITH ";TITLE$(PLACED,PLACED)
7230 FOR A=1 TO 500:NEXT A
7240 PEN ON
7250 IF PEN(3)=0 THEN GOTO 7240
7260 X1=PEN(4):Y1=PEN(5)
7270 PSET (X1,Y1)
7280 LOCATE 21,1:PRINT SPC(39)
7290 LOCATE 22,1:PRINT SPC(39)
7300 LOCATE 23,1:PRINT "CORRECT POINT?     YES   NO     "
7310 LINE (150,185)-(175,170),1,B:LINE (195,185)-(220,170),1,B
7320 PEN ON
7330 IF PEN(3)=0 THEN GOTO 7320
7340 IF PEN(8)>21 AND PEN(9)>24 THEN PSET (X1,Y1),0:GOTO 7230 ELSE GOTO 7350
7350 PSET (X1,Y1),0
7360 IF Y1>=PINPTS(1,2)-3 AND Y1<=PINPTS(1,2)+5 AND X1>PINPTS(1,1) THEN
      XDIST=X1-PLANPTS(ACTIVITY,1,2):YDIST=PINPTS(1,2)-PLANPTS(ACTIVITY,1,3):
      SPACE(ACTIVITY,6)=1:GOTO 7440 ELSE GOTO 7370
7370 IF Y1>=PINPTS(4,2)-5 AND Y1<=PINPTS(4,2)+3 AND X1<PINPTS(3,1) THEN
      XDIST=X1-PLANPTS(ACTIVITY,4,2):YDIST=PINPTS(4,2)-PLANPTS(ACTIVITY,4,3):
      SPACE(ACTIVITY,6)=3:GOTO 7440 ELSE GOTO 7380
7380 IF X1>=PINPTS(2,1)-5 AND X1<=PINPTS(2,1)+3 AND Y1>PINPTS(2,2)+5 THEN
      XDIST=PINPTS(2,1)-PLANPTS(ACTIVITY,2,2):YDIST=Y1-PLANPTS(ACTIVITY,2,3):
      SPACE(ACTIVITY,6)=2:GOTO 7440 ELSE GOTO 7390
7390 IF X1>=PINPTS(1,1)-3 AND X1<=PINPTS(1,1)+5 AND Y1<PINPTS(4,2)-5 THEN
      XDIST=PINPTS(1,1)-PLANPTS(ACTIVITY,1,2):YDIST=Y1-PLANPTS(ACTIVITY,1,3):
      SPACE(ACTIVITY,6)=4:GOTO 7440 ELSE GOTO 7400
7400 IF Y1>=HALLPTS(2,2)-5 AND Y1<=HALLPTS(2,2)+2 AND X1>PINPTS(1,1) THEN
      XDIST=X1-PLANPTS(ACTIVITY,4,2):YDIST=HALLPTS(2,2)-PLANPTS(ACTIVITY,4,3):
      SPACE(ACTIVITY,6)=1:GOTO 7440 ELSE GOTO 7410
7410 IF Y1>=HALLPTS(4,2)-2 AND Y1<=HALLPTS(4,2)+5 AND X1<PINPTS(3,1) THEN
      XDIST=X1-PLANPTS(ACTIVITY,1,2):YDIST=HALLPTS(4,2)-PLANPTS(ACTIVITY,1,3):
      SPACE(ACTIVITY,6)=3:GOTO 7440 ELSE GOTO 7420
7420 IF X1>=HALLPTS(3,1)-2 AND X1<=HALLPTS(3,1)+5 AND Y1>PINPTS(2,2)+5 THEN
      XDIST=HALLPTS(3,1)-PLANPTS(ACTIVITY,1,2):YDIST=Y1-PLANPTS(ACTIVITY,1,3):
      SPACE(ACTIVITY,6)=2:GOTO 7440 ELSE GOTO 7430
7430 IF X1>=HALLPTS(1,1)-5 AND X1<=HALLPTS(1,1)+2 AND Y1< PINPTS(4,2)-5 THEN
      XDIST=HALLPTS(1,1)-PLANPTS(ACTIVITY,2,2):YDIST=Y1-PLANPTS(ACTIVITY,2,3):
      SPACE(ACTIVITY,6)=4:GOTO 7440 ELSE STOP
7440 '
7450 FLAG=1:I=1:COUNT=0
7460 WHILE FLAG
7470     I=PLANPTS(ACTIVITY,I,1)
```

```
7480      LINE (PLANPTS(ACTIVITY,PLANPTS(ACTIVITY,I,1),2),PLANPTS(ACTIVITY,
            PLANPTS(ACTIVITY,I,1),3))-(PLANPTS(ACTIVITY,PLANPTS(ACTIVITY,I,4),2)
            ,PLANPTS(ACTIVITY,PLANPTS(ACTIVITY,I,4),3)),0
7490      I=PLANPTS(ACTIVITY,I,4)
7500      COUNT=COUNT+1
7510      IF COUNT>=SPACE(ACTIVITY,11) THEN FLAG=0
7520 WEND
7530 IFLAG=1:GOTO 1020
7540 '
7550 'Modify location of activity
7560 '
7570 CFLAG=0
7580 LOCATE 23,1:PRINT "ANY CHANGES?        YES    NO      "
7590 LINE (150,185)-(175,170),1,B:LINE (195,185)-(220,170),1,B
7600 PEN ON
7610 IF PEN(3)=0 THEN GOTO 7600
7620 IF PEN(8)>21 AND PEN(9)>24 THEN GOTO 2510
7630 LINE (150,185)-(175,170),0,B:LINE (195,185)-(220,170),0,B
7640 LOCATE 23,1:PRINT "  MODIFY    REDRAW                  "
7650 LINE (5,185)-(75,170),1,B:LINE (80,185)-(150,170),1,B
7660 PEN ON
7670 IF PEN(3)=0 THEN GOTO 7660
7680 IF PEN(8)>21 AND PEN(9)>10 THEN GOTO 8630
7690 LINE (5,185)-(75,170),0,B:LINE (80,185)-(150,170),0,B
7700 LOCATE 23,1:PRINT SPC(39)
7710 LOCATE 21,1:PRINT "INDICATE NEW ACTIVITY LOCATION"
7720 FOR A=1 TO 500:NEXT A
7730 PEN ON
7740 IF PEN(3)=0 THEN GOTO 7730
7750 X1=PEN(4):Y1=PEN(5)
7760 PSET (X1,Y1)
7770 LOCATE 23,1:PRINT "CORRECT POINT?      YES    NO      "
7780 LINE (150,185)-(175,170),1,B:LINE (195,185)-(220,170),1,B
7790 PEN ON
7800 IF PEN(3)=0 THEN GOTO 7790
7810 IF PEN(8)>21 AND PEN(9)>24 THEN PSET (X1,Y1),0:GOTO 7720 ELSE GOTO 7820
7820 PSET (X1,Y1),0
7830 '
7840 'Check TRACK array to determine activity to be moved
7850 '
7860 FOR A=1 TO N
7870  IF X1>=PLANPTS(A,1,2) AND X1<=PLANPTS(A,2,2) AND Y1>=PLANPTS(A,1,3)
        AND Y1<=PLANPTS(A,4,3) THEN ACTIVITY=A:GOTO 7900 ELSE GOTO 7880
7880 NEXT A
7890 STOP
7900 FOR A=1 TO NPLACED
7910    IF ACTIVITY=TRACK(A,1) THEN GOTO 7940
7920 NEXT A
7930 STOP
7940 LOCATE 1,38:PRINT "*"
7950 FOR B=1 TO 500:NEXT B
7960 PEN ON
7970 IF PEN(3)=0 THEN GOTO 7960
7980 X2=PEN(4):Y2=PEN(5)
7990 PSET (X2,Y2)
8000 FOR B=1 TO 500:NEXT B
8010 PEN ON
8020 IF PEN(3)=0 THEN GOTO 8010
8030 IF PEN(8)>21 AND PEN(9)>24 THEN PSET (X2,Y2),0:GOTO 7950 ELSE GOTO 8040
8040 PSET (X2,Y2),0
```

```
8050 LOCATE 1,38:PRINT "  "
8060 '
8070 'North side pinwheel check (south)
8080 '
8090 IF Y2>=PINPTS(1,2)-3 AND Y2<=PINPTS(1,2)+5 AND X2>PINPTS(1,1) THEN
      XDIST=X2-PLANPTS(ACTIVITY,1,2):YDIST=PINPTS(1,2)-PLANPTS(ACTIVITY,1,3):
      SPACE(ACTIVITY,6)=1:GOTO 8390 ELSE GOTO 8100
8100 '
8110 'South side pinwheel check (north)
8120 '
8130 IF Y2>=PINPTS(4,2)-5 AND Y2<=PINPTS(4,2)+3 AND X2<PINPTS(3,1) THEN
      XDIST=X2-PLANPTS(ACTIVITY,4,2):YDIST=PINPTS(4,2)-PLANPTS(ACTIVITY,4,3):
      SPACE(ACTIVITY,6)=3:GOTO 8390 ELSE GOTO 8140
8140 '
8150 'East side pinwheel check (west)
8160 '
8170 IF X2>=PINPTS(2,1)-5 AND X2<=PINPTS(2,1)+3 AND Y2>PINPTS(2,2)+5 THEN
      XDIST=PINPTS(2,1)-PLANPTS(ACTIVITY,2,2):YDIST=Y2-PLANPTS(ACTIVITY,2,3):
      SPACE(ACTIVITY,6)=2:GOTO 8390 ELSE GOTO 8180
8180 '
8190 'West side pinwheel check (east)
8200 '
8210 IF X2>=PINPTS(1,1)-3 AND X2<=PINPTS(1,1)+5 AND Y2<PINPTS(4,2)-5 THEN
      XDIST=PINPTS(1,1)-PLANPTS(ACTIVITY,1,2):YDIST=Y2-PLANPTS(ACTIVITY,1,3):
      SPACE(ACTIVITY,6)=4:GOTO 8390 ELSE GOTO 8220
8220 '
8230 'North side pinwheel check (north)
8240 '
8250 IF Y2>=HALLPTS(2,2)-5 AND Y2<=HALLPTS(2,2)+2 AND X2>PINPTS(1,1) THEN
      XDIST=X2-PLANPTS(ACTIVITY,4,2):YDIST=HALLPTS(2,2)-PLANPTS(ACTIVITY,4,3):
      SPACE(ACTIVITY,6)=1:GOTO 8390 ELSE GOTO 8260
8260 '
8270 'South side pinwheel check (south)
8280 '
8290 IF Y2>=HALLPTS(4,2)-2 AND Y2<=HALLPTS(4,2)+5 AND X2<PINPTS(3,1) THEN
      XDIST=X2-PLANPTS(ACTIVITY,1,2):YDIST=HALLPTS(4,2)-PLANPTS(ACTIVITY,1,3):
      SPACE(ACTIVITY,6)=3:GOTO 8390 ELSE GOTO 8300
8300 '
8310 'East side pinwheel check (east)
8320 '
8330 IF X2>=HALLPTS(3,1)-2 AND X2<=HALLPTS(3,1)+5 AND Y2>PINPTS(2,2)+5 THEN
      XDIST=HALLPTS(3,1)-PLANPTS(ACTIVITY,1,2):YDIST=Y2-PLANPTS(ACTIVITY,1,3):
      SPACE(ACTIVITY,6)=2:GOTO 8390 ELSE GOTO 8340
8340 '
8350 'West side pinwheel check (west)
8360 '
8370 IF X2>=HALLPTS(1,1)-5 AND X2<=HALLPTS(1,1)+2 AND Y2< PINPTS(4,2)-5 THEN
      XDIST=HALLPTS(1,1)-PLANPTS(ACTIVITY,2,2):YDIST=Y2-PLANPTS(ACTIVITY,2,3):
      SPACE(ACTIVITY,6)=4:GOTO 8390 ELSE GOTO 8380
8380 STOP
8390 '
8400 'Erase existing activity
8410 '
8420 XPLACE=INT(PLANPTS(ACTIVITY,1,2)/8):YPLACE=INT(PLANPTS(ACTIVITY,1,3)/8)
8430 IF TRACK(A,3)=PINPTS(1,2) AND TRACK(A,5)=PINPTS(1,2) THEN
      LOCATE YPLACE+1,XPLACE+1:GOTO 8520 ELSE GOTO 8440
8440 IF TRACK(A,4)=PINPTS(2,1) AND TRACK(A,6)=PINPTS(2,1) THEN
      LOCATE YPLACE+1,XPLACE+2:GOTO 8520 ELSE GOTO 8450
8450 IF TRACK(A,7)=PINPTS(3,2) AND TRACK(A,9)=PINPTS(3,2) THEN
      LOCATE YPLACE+1,XPLACE+2:GOTO 8520 ELSE GOTO 8460
```

```
8460 IF TRACK(A,8)=PINPTS(4,1) AND TRACK(A,2)=PINPTS(4,1) THEN
      LOCATE YPLACE+1,XPLACE+1:GOTO 8520 ELSE GOTO 8470
8470 IF TRACK(A,9)=HALLPTS(2,2) AND TRACK(A,7)=HALLPTS(2,2) THEN
      LOCATE YPLACE-1,XPLACE+1:GOTO 8520 ELSE GOTO 8480
8480 IF TRACK(A,2)=HALLPTS(3,1) AND TRACK(A,8)=HALLPTS(3,1) THEN
      LOCATE YPLACE,XPLACE+5  :GOTO 8520 ELSE GOTO 8490
8490 IF TRACK(A,3)=HALLPTS(4,2) AND TRACK(A,5)=HALLPTS(4,2) THEN
      LOCATE YPLACE+5,XPLACE+1:GOTO 8520 ELSE GOTO 8500
8500 IF TRACK(A,4)=HALLPTS(1,1) AND TRACK(A,6)=HALLPTS(1,1) THEN
      LOCATE YPLACE,XPLACE-1  :GOTO 8520 ELSE GOTO 8510
8510 STOP
8520 IF ACTIVITY< 10 THEN PRINT " "
8530 IF ACTIVITY>=10 THEN PRINT "  "
8540 FLAG=1:I=1:COUNT=0
8550 WHILE FLAG
8560      I=PLANPTS(ACTIVITY,I,1)
8570      LINE (PLANPTS(ACTIVITY,PLANPTS(ACTIVITY,I,1),2),PLANPTS(ACTIVITY,
            PLANPTS(ACTIVITY,I,1),3))-(PLANPTS(ACTIVITY,PLANPTS(ACTIVITY,I,4),2)
            ,PLANPTS(ACTIVITY,PLANPTS(ACTIVITY,I,4),3)),0
8580      I=PLANPTS(ACTIVITY,I,4)
8590      COUNT=COUNT+1
8600      IF COUNT>=SPACE(ACTIVITY,11) THEN FLAG=0
8610 WEND
8620 CFLAG=1:GOTO 1020
8630 '
8640 'Redraw screen
8650 '
8660 LINE (5,185)-(75,170),0,B:LINE (80,185)-(150,170),0,B
8670 LOCATE 23,1:PRINT SPC(39)
8680 FOR A=1 TO NPLACED
8690 FLAG=1:I=1:COUNT=0
8700 WHILE FLAG
8710      I=PLANPTS(TRACK(A,1),I,1)
8720      LINE (PLANPTS(TRACK(A,1),PLANPTS(TRACK(A,1),I,1),2),PLANPTS(TRACK(A,1),
          PLANPTS(TRACK(A,1),I,1),3))-(PLANPTS(TRACK(A,1),PLANPTS(TRACK(A,1),I,4)
          ,2),PLANPTS(TRACK(A,1),PLANPTS(TRACK(A,1),I,4),3)),1
8730      I=PLANPTS(TRACK(A,1),I,4)
8740      COUNT=COUNT+1
8750      IF COUNT>=SPACE(TRACK(A,1),11) THEN FLAG=0
8760 WEND
8770 NEXT A
8780 LINE (PINPTS(1,1),PINPTS(1,2))-(PINPTS(3,1),PINPTS(3,2)),1,B
8790 LINE (HALLPTS(1,1),HALLPTS(1,2))-(PINPTS(4,1),PINPTS(4,2)),1,B
8800 LINE (HALLPTS(2,1),HALLPTS(2,2))-(PINPTS(1,1),PINPTS(1,2)),1,B
8810 LINE (HALLPTS(3,1),HALLPTS(3,2))-(PINPTS(2,1),PINPTS(2,2)),1,B
8820 LINE (HALLPTS(4,1),HALLPTS(4,2))-(PINPTS(3,1),PINPTS(3,2)),1,B
8830 GOTO 7540
8840 '
8850 'Allow another circulation design to be developed
8860 '
8870 LOCATE 21,1:PRINT SPC(39)
8880 LINE (150,185)-(175,170),0,B:LINE (195,185)-(220,170),0,B
8890 LOCATE 23,1:PRINT SPC(39)
8900 LOCATE 23,1:PRINT "SAVE PLAN?          YES   NO      "
8910 LINE (150,185)-(175,170),1,B:LINE (195,185)-(220,170),1,B
8920 PEN ON
8930 IF PEN(3)=0 THEN GOTO 8920
8940 IF PEN(8)>21 AND PEN(9)<23 THEN JUMP$="RADIAL":GOTO 9210 ELSE GOTO 8950
8950 LOCATE 23,1:PRINT "PLACE ON SITE?     YES   NO      "
8960 LINE (150,185)-(175,170),1,B:LINE (195,185)-(220,170),1,B
```

```
8970 PEN ON
8980 IF PEN(3)=0 THEN GOTO 8970
8990 IF PEN(8)>21 AND PEN(9)<23 THEN GOTO 9200
9000 LOCATE 23,1:PRINT "ANOTHER DESIGN?    YES   NO      "
9010 LINE (150,185)-(175,170),1,B:LINE (195,185)-(220,170),1,B
9020 PEN ON
9030 IF PEN(3)=0 THEN GOTO 9020
9040 IF PEN(8)>21 AND PEN(9)>24 THEN GOTO 9120
9050 LINE (150,185)-(175,170),0,B:LINE (195,185)-(220,170),0,B
9060 LOCATE 23,1:PRINT "UPDATE DATA?       YES   NO      "
9070 LINE (150,185)-(175,170),1,B:LINE (195,185)-(220,170),1,B
9080 PEN ON
9090 IF PEN(3)=0 THEN GOTO 9080
9100 IF PEN(8)>21 AND PEN(9)>24 THEN GOTO 9190
9110 GOTO 9180
9120 LOCATE 23,1:PRINT "START AGAIN?       YES   NO      "
9130 LINE (150,185)-(175,170),1,B:LINE (195,185)-(220,170),1,B
9140 PEN ON
9150 IF PEN(3)=0 THEN GOTO 9140
9160 IF PEN(8)>21 AND PEN(9)<23 THEN GOTO 9170 ELSE STOP
9170 CHAIN MERGE "CREATE",10,DELETE 10-9160
9180 CHAIN MERGE "SPACE",10,ALL,DELETE 10-9170
9190 CHAIN MERGE "GEOMTRY",10,ALL,DELETE 10-9180
9200 CHAIN MERGE "SITE",10,ALL,DELETE 10-9190
9210 CHAIN MERGE "PLNFILE",10,ALL,DELETE 10-9200
```

```
10 '*************************************************************************
20 'CLUSTER Subroutine
30 '*************************************************************************
40 'This routine allows the user to determine his own circulation pattern.
50 'CREATE selects the order of entry and prompts the user to place the
60 'activity at a grid coordinate.
70 '
80 CLS:PEN OFF
90 JUMP=4
100 LOCATE 2,12:PRINT "CLUSTER Subroutine"
110 LOCATE 23,1:INPUT "ENTER GRID SIZE  ",GRID
120 GRID=GRID*SCALE
130 CLS
140 FOR A=100 TO 220 STEP GRID
150     LINE (A,10)-(A,90),1,,&HAAAA
160 NEXT A
170 FOR B=10 TO 90 STEP GRID
180     LINE (100,B)-(220,B),1,,&HAAAA
190 NEXT B
200 '
210 'Determine activity with largest square footage
220 '
230 MAXAREA=0:NPLACED=0
240 FOR A=1 TO N
250     IF SPACE(A,1)>MAXAREA THEN ACTIVITY=A:MAXAREA=SPACE(A,1) ELSE GOTO 260
260 NEXT A
270 '
280 'Display activity to be placed next
290 '
300 XDIST=250-PLANPTS(ACTIVITY,1,2)
310 YDIST=150-PLANPTS(ACTIVITY,1,3)
320 FOR A=1 TO SPACE(ACTIVITY,11)
330     PLANPTS(ACTIVITY,A,2)=PLANPTS(ACTIVITY,A,2)+XDIST
340     PLANPTS(ACTIVITY,A,3)=PLANPTS(ACTIVITY,A,3)+YDIST
350 NEXT A
360 FLAG=1:I=1:COUNT=0
370 NPLACED=NPLACED+1
380 TRACK(NPLACED,1)=ACTIVITY
390 WHILE FLAG
400     I=PLANPTS(ACTIVITY,I,1)
410     LINE (PLANPTS(ACTIVITY,PLANPTS(ACTIVITY,I,1),2),PLANPTS(ACTIVITY,
          PLANPTS(ACTIVITY,I,1),3))-(PLANPTS(ACTIVITY,PLANPTS(ACTIVITY,I,4),2)
          ,PLANPTS(ACTIVITY,PLANPTS(ACTIVITY,I,4),3)),1
420     I=PLANPTS(ACTIVITY,I,4)
430     COUNT=COUNT+1
440     IF COUNT>=SPACE(ACTIVITY,11) THEN FLAG=0
450 WEND
460 LOCATE 20,1:PRINT TITLE$(ACTIVITY,ACTIVITY)
470 IF SPACE(ACTIVITY,6)=0 THEN WALL$="      "
480 IF SPACE(ACTIVITY,6)=1 THEN WALL$="NORTH"
490 IF SPACE(ACTIVITY,6)=2 THEN WALL$="EAST "
500 IF SPACE(ACTIVITY,6)=3 THEN WALL$="SOUTH"
510 IF SPACE(ACTIVITY,6)=4 THEN WALL$="WEST "
520 LOCATE 21,1:PRINT "WALL ASSIGNMENT:  ";WALL$
530 '
540 'Locate activity on axis
550 '
560 FOR A=1 TO 500:NEXT A
570 PEN ON
580 IF PEN(3)=0 THEN GOTO 570
```

```
590 X1=PEN(4):Y1=PEN(5)
600 PSET (X1,Y1)
610 LOCATE 21,1:PRINT SPC(30)
620 LOCATE 22,1:PRINT SPC(30)
630 LOCATE 23,1:PRINT "CORRECT POINT?      YES   NO      "
640 LINE (150,185)-(175,170),1,B:LINE (195,185)-(220,170),1,B
650 PEN ON
660 IF PEN(3)=0 THEN GOTO 650
670 IF PEN(8)>21 AND PEN(9)>24 THEN PSET (X1,Y1),0:GOTO 560 ELSE GOTO 680
680 PSET (X1,Y1),0
690 FLAG=1:I=1:COUNT=0
700 WHILE FLAG
710     I=PLANPTS(ACTIVITY,I,1)
720     LINE (PLANPTS(ACTIVITY,PLANPTS(ACTIVITY,I,1),2),PLANPTS(ACTIVITY,
          PLANPTS(ACTIVITY,I,1),3))-(PLANPTS(ACTIVITY,PLANPTS(ACTIVITY,I,4),2)
          ,PLANPTS(ACTIVITY,PLANPTS(ACTIVITY,I,4),3)),0
730     I=PLANPTS(ACTIVITY,I,4)
740     COUNT=COUNT+1
750     IF COUNT>=SPACE(ACTIVITY,11) THEN FLAG=0
760 WEND
770 XDIST=X1-PLANPTS(ACTIVITY,1,2):YDIST=Y1-PLANPTS(ACTIVITY,1,3)
780 FOR A=1 TO SPACE(ACTIVITY,11)
790     PLANPTS(ACTIVITY,A,2)=PLANPTS(ACTIVITY,A,2)+XDIST
800     PLANPTS(ACTIVITY,A,3)=PLANPTS(ACTIVITY,A,3)+YDIST
810 NEXT A
820 FLAG=1:I=1:COUNT=0
830 WHILE FLAG
840     I=PLANPTS(ACTIVITY,I,1)
850     LINE (PLANPTS(ACTIVITY,PLANPTS(ACTIVITY,I,1),2),PLANPTS(ACTIVITY,
          PLANPTS(ACTIVITY,I,1),3))-(PLANPTS(ACTIVITY,PLANPTS(ACTIVITY,I,4),2)
          ,PLANPTS(ACTIVITY,PLANPTS(ACTIVITY,I,4),3)),1
860     I=PLANPTS(ACTIVITY,I,4)
870     COUNT=COUNT+1
880     IF COUNT>=SPACE(ACTIVITY,11) THEN FLAG=0
890 WEND
900 XPLACE=INT(PLANPTS(ACTIVITY,1,2)/8):YPLACE=INT(PLANPTS(ACTIVITY,1,3)/8)
910 LOCATE YPLACE,XPLACE+1
920 IF ACTIVITY< 10 THEN PRINT USING "#";ACTIVITY
930 IF ACTIVITY>=10 THEN PRINT USING "##";ACTIVITY
940 IF NPLACED<= 6 THEN GOTO 970
950 IF NPLACED<=12 THEN GOTO 1000
960 IF NPLACED<=20 THEN GOTO 1030
970 LOCATE 12+NPLACED,1:PRINT USING "#";ACTIVITY
980 LOCATE 12+NPLACED,3:PRINT USING "\         \";TITLE$(ACTIVITY,ACTIVITY)
990 GOTO 1050
1000 LOCATE 6+NPLACED,14:PRINT USING "#";ACTIVITY
1010 LOCATE 6+NPLACED,17:PRINT USING "\         \";TITLE$(ACTIVITY,ACTIVITY)
1020 GOTO 1050
1030 LOCATE NPLACED,28:PRINT USING "#";ACTIVITY
1040 LOCATE NPLACED,31:PRINT USING "\         \";TITLE$(ACTIVITY,ACTIVITY)
1050 '
1060 'Allow changes to be made in design
1070 '
1080 LOCATE 20,1:PRINT SPC(39)
1090 LOCATE 21,1:PRINT SPC(39)
1100 LOCATE 22,1:PRINT SPC(39)
1110 LOCATE 23,1:PRINT "ANY CHANGES?        YES   NO     "
1120 LINE (150,185)-(175,170),1,B:LINE (195,185)-(220,170),1,B
1130 PEN ON
1140 IF PEN(3)=0 THEN GOTO 1130
```

```
1150 IF PEN(8)>21 AND PEN(9)>24 THEN GOTO 1160 ELSE GOTO 1770
1160 LINE (150,185)-(175,170),0,B:LINE (195,185)-(220,170),0,B
1170 LOCATE 23,1:PRINT SPC(39)
1180 IF NPLACED=N THEN GOTO 2710
1190 '
1200 'Select next activity to be placed
1210 '
1220 MAXREL=0:MINREL=-30:XFLAG=0
1230  FOR A=1 TO NPLACED
1240    FOR B=1 TO N
1250      FOR C=1 TO NPLACED
1260        IF B=TRACK(C,1) THEN GOTO 1300
1270      NEXT C
1280      IF TITLE(TRACK(A,1),B)>MAXREL THEN MAXREL=TITLE(TRACK(A,1),B):
           ACTIVITY=B:PLACED=A ELSE GOTO 1290
1290      IF TITLE(TRACK(A,1),B)=MINREL THEN XFLAG=1:TEMPACTIVITY=B:
           TEMPPLACED=A ELSE GOTO 1300
1300    NEXT B
1310 NEXT A
1320 IF XFLAG=1 THEN ACTIVITY=TEMPACTIVITY:PLACED=TEMPPLACED:GOTO 1440
      ELSE GOTO 1330
1330 IF MAXREL>17 THEN GOTO 1440
1340 '
1350 'No A, B, or X relationships, select largest area
1360 '
1370 MAXAREA=0
1380 FOR C=1 TO N
1390    FOR D=1 TO NPLACED
1400      IF C=TRACK(D,1) THEN GOTO 1430
1410    NEXT D
1420    IF SPACE(C,1)>MAXAREA THEN ACTIVITY=C:MAXAREA=SPACE(C,1)
         ELSE GOTO 1430
1430 NEXT C
1440 '
1450 'Display incoming activity and relationships
1460 '
1470 XDIST=250-PLANPTS(ACTIVITY,1,2)
1480 YDIST=150-PLANPTS(ACTIVITY,1,3)
1490 FOR D=1 TO SPACE(ACTIVITY,11)
1500    PLANPTS(ACTIVITY,D,2)=PLANPTS(ACTIVITY,D,2)+XDIST
1510    PLANPTS(ACTIVITY,D,3)=PLANPTS(ACTIVITY,D,3)+YDIST
1520 NEXT D
1530 FLAG=1:I=1:COUNT=0
1540 NPLACED=NPLACED+1
1550 TRACK(NPLACED,1)=ACTIVITY
1560 WHILE FLAG
1570     I=PLANPTS(ACTIVITY,I,1)
1580     LINE (PLANPTS(ACTIVITY,PLANPTS(ACTIVITY,I,1),2),PLANPTS(ACTIVITY,
           PLANPTS(ACTIVITY,I,1),3))-(PLANPTS(ACTIVITY,PLANPTS(ACTIVITY,I,4),2)
           ,PLANPTS(ACTIVITY,PLANPTS(ACTIVITY,I,4),3)),1
1590     I=PLANPTS(ACTIVITY,I,4)
1600     COUNT=COUNT+1
1610     IF COUNT>=SPACE(ACTIVITY,11) THEN FLAG=0
1620 WEND
1630 LOCATE 20,1:PRINT TITLE$(ACTIVITY,ACTIVITY)
1640 IF SPACE(ACTIVITY,6)=0 THEN WALL$="      "
1650 IF SPACE(ACTIVITY,6)=1 THEN WALL$="NORTH"
1660 IF SPACE(ACTIVITY,6)=2 THEN WALL$="EAST "
1670 IF SPACE(ACTIVITY,6)=3 THEN WALL$="SOUTH"
1680 IF SPACE(ACTIVITY,6)=4 THEN WALL$="WEST "
```

```
1690 LOCATE 21,1:PRINT "WALL ASSIGNMENT:  ";WALL$
1700 IF XFLAG=1 THEN GOTO 1710 ELSE GOTO 1730
1710 LOCATE 22,1:PRINT "X-RELATIONSHIP WITH ";TRACK(PLACED,1)
1720 GOTO 1760
1730 IF MAXREL>17 THEN GOTO 1740 ELSE GOTO 1760
1740 LOCATE 22,1:PRINT TITLE$(ACTIVITY,TRACK(PLACED,1))
1750 LOCATE 22,2:PRINT "-RELATIONSHIP WITH ";ACTIVITY
1760 GOTO 530
1770 '
1780 'Modify or redraw activity
1790 '
1800 LINE (150,185)-(175,170),0,B:LINE (195,185)-(220,170),0,B
1810 LOCATE 23,1:PRINT SPC(39)
1820 LOCATE 23,1:PRINT "  MODIFY   REDRAW                "
1830 LINE (5,185)-(75,170),1,B:LINE (80,185)-(150,170),1,B
1840 PEN ON
1850 IF PEN(3)=0 THEN GOTO 1840
1860 IF PEN(8)>21 AND PEN(9)>10 THEN GOTO 2540
1870 LINE (5,185)-(75,170),0,B:LINE (80,185)-(150,170),0,B
1880 LOCATE 23,1:PRINT SPC(39)
1890 LOCATE 21,1:PRINT "INDICATE NEW ACTIVITY LOCATION  "
1900 FOR A=1 TO 500:NEXT A
1910 PEN ON
1920 IF PEN(3)=0 THEN GOTO 1910
1930 X1=PEN(4):Y1=PEN(5)
1940 PSET (X1,Y1)
1950 LOCATE 23,1:PRINT "CORRECT POINT?     YES   NO      "
1960 LINE (150,185)-(175,170),1,B:LINE (195,185)-(220,170),1,B
1970 PEN ON
1980 IF PEN(3)=0 THEN GOTO 1970
1990 IF PEN(8)>21 AND PEN(9)>24 THEN PSET (X1,Y1),0:GOTO 1900 ELSE GOTO 2000
2000 PSET (X1,Y1),0
2010 '
2020 'Determine which activity to be moved
2030 '
2040 FOR A=1 TO N
2050    IF X1>=PLANPTS(A,1,2) AND X1<=PLANPTS(A,2,2) AND Y1>=PLANPTS(A,1,3)
         AND Y1<=PLANPTS(A,4,3) THEN ACTIVITY=A:GOTO 2080 ELSE GOTO 2060
2060 NEXT A
2070 STOP
2080 LOCATE 1,38:PRINT "*"
2090 FOR B=1 TO 500:NEXT B
2100 PEN ON
2110 IF PEN(3)=0 THEN GOTO 2100
2120 X2=PEN(4):Y2=PEN(5)
2130 PSET (X2,Y2)
2140 FOR B=1 TO 500:NEXT B
2150 PEN ON
2160 IF PEN(3)=0 THEN GOTO 2150
2170 IF PEN(8)>21 AND PEN(9)>24 THEN PSET (X2,Y2),0:GOTO 2090 ELSE GOTO 2180
2180 PSET (X2,Y2),0
2190 LOCATE 1,38:PRINT "  "
2200 XDIST=X2-PLANPTS(ACTIVITY,1,2):YDIST=Y2-PLANPTS(ACTIVITY,1,3)
2210 '
2220 'Erase existing activity
2230 '
2240 XPLACE=INT(PLANPTS(ACTIVITY,1,2)/8):YPLACED=INT(PLANPTS(ACTIVITY,1,3)/8)
2250 LOCATE YPLACE,XPLACE+1:PRINT "  "
2260 FLAG=1:I=1:COUNT=0
2270 WHILE FLAG
```

```
2280      I=PLANPTS(ACTIVITY,I,1)
2290      LINE (PLANPTS(ACTIVITY,PLANPTS(ACTIVITY,I,1),2),PLANPTS(ACTIVITY,
            PLANPTS(ACTIVITY,I,1),3))-(PLANPTS(ACTIVITY,PLANPTS(ACTIVITY,I,4),2)
            ,PLANPTS(ACTIVITY,PLANPTS(ACTIVITY,I,4),3)),0
2300      I=PLANPTS(ACTIVITY,I,4)
2310      COUNT=COUNT+1
2320      IF COUNT>=SPACE(ACTIVITY,11) THEN FLAG=0
2330 WEND
2340 '
2350 'Locate modified activity on design
2360 '
2370 FOR B=1 TO SPACE(ACTIVITY,11)
2380    PLANPTS(ACTIVITY,B,2)=PLANPTS(ACTIVITY,B,2)+XDIST
2390    PLANPTS(ACTIVITY,B,3)=PLANPTS(ACTIVITY,B,3)+YDIST
2400 NEXT B
2410 FLAG=1:I=1:COUNT=0
2420 WHILE FLAG
2430      I=PLANPTS(ACTIVITY,I,1)
2440      LINE (PLANPTS(ACTIVITY,PLANPTS(ACTIVITY,I,1),2),PLANPTS(ACTIVITY,
            PLANPTS(ACTIVITY,I,1),3))-(PLANPTS(ACTIVITY,PLANPTS(ACTIVITY,I,4),2)
            ,PLANPTS(ACTIVITY,PLANPTS(ACTIVITY,I,4),3)),1
2450      I=PLANPTS(ACTIVITY,I,4)
2460      COUNT=COUNT+1
2470      IF COUNT>=SPACE(ACTIVITY,11) THEN FLAG=0
2480 WEND
2490 XPLACE=INT(PLANPTS(ACTIVITY,1,2)/8):YPLACE=INT(PLANPTS(ACTIVITY,1,3)/8)
2500 LOCATE YPLACE,XPLACE+1
2510 IF ACTIVITY< 10 THEN PRINT USING "#";ACTIVITY
2520 IF ACTIVITY>=10 THEN PRINT USING "##";ACTIVITY
2530 GOTO 1050
2540 '
2550 'Redraw screen
2560 '
2570 LOCATE 21,1:PRINT SPC(39)
2580 LINE (5,185)-(75,170),0,B:LINE (80,185)-(150,170),0,B
2590 LOCATE 23,1:PRINT SPC(39)
2600 FOR A=1 TO NPLACED
2610 FLAG=1:I=1:COUNT=0
2620 WHILE FLAG
2630      I=PLANPTS(TRACK(A,1),I,1)
2640      LINE (PLANPTS(TRACK(A,1),PLANPTS(TRACK(A,1),I,1),2),PLANPTS(TRACK(A,1),
            PLANPTS(TRACK(A,1),I,1),3))-(PLANPTS(TRACK(A,1),PLANPTS(TRACK(A,1),I,4
            ),2),PLANPTS(TRACK(A,1),PLANPTS(TRACK(A,1),I,4),3)),1
2650      I=PLANPTS(TRACK(A,1),I,4)
2660      COUNT=COUNT+1
2670      IF COUNT>=SPACE(TRACK(A,1),11) THEN FLAG=0
2680 WEND
2690 NEXT A
2700 GOTO 1050
2710 '
2720 'Allow another circulation design to be developed
2730 '
2740 LOCATE 20,1:PRINT SPC(39)
2750 LINE (150,185)-(175,170),0,B:LINE (195,185)-(220,170),0,B
2760 LOCATE 23,1:PRINT SPC(39)
2770 LOCATE 23,1:PRINT "ERASE GRID?         YES    NO      "
2780 LINE (150,185)-(175,170),1,B:LINE (195,185)-(220,170),1,B
2790 PEN ON
2800 IF PEN(3)=0 THEN GOTO 2790
2810 IF PEN(8)>21 AND PEN(9)>24 THEN GOTO 2980
```

```
2820 FOR A=100 TO 220 STEP GRID
2830    LINE (A,10)-(A,90),0,,&HAAAA
2840 NEXT A
2850 FOR B=10 TO 90 STEP GRID
2860    LINE (100,B)-(220,B),0,,&HAAAA
2870 NEXT B
2880 FOR A=1 TO NPLACED
2890 FLAG=1:I=1:COUNT=0
2900 WHILE FLAG
2910     I=PLANPTS(TRACK(A,1),I,1)
2920     LINE (PLANPTS(TRACK(A,1),PLANPTS(TRACK(A,1),I,1),2),PLANPTS(TRACK(A,1),
          PLANPTS(TRACK(A,1),I,1),3))-(PLANPTS(TRACK(A,1),PLANPTS(TRACK(A,1),I,4
          ),2),PLANPTS(TRACK(A,1),PLANPTS(TRACK(A,1),I,4),3)),1
2930     I=PLANPTS(TRACK(A,1),I,4)
2940     COUNT=COUNT+1
2950     IF COUNT>=SPACE(TRACK(A,1),11) THEN FLAG=0
2960 WEND
2970 NEXT A
2980 LINE (150,185)-(175,170),0,B:LINE (195,185)-(175,170),0,B
2990 LOCATE 23,1:PRINT SPC(39)
3000 LOCATE 23,1:PRINT "SAVE PLAN?          YES    NO      "
3010 LINE (150,185)-(175,170),1,B:LINE (195,185)-(220,170),1,B
3020 PEN ON
3030 IF PEN(3)=0 THEN GOTO 3020
3040 IF PEN(8)>21 AND PEN(9)<23 THEN JUMP$="CLUSTER":GOTO 3300 ELSE GOTO 3050
3050 LOCATE 23,1:PRINT "PLACE ON SITE?      YES    NO      "
3060 LINE (150,185)-(175,170),1,B:LINE (195,185)-(220,170),1,B
3070 PEN ON
3080 IF PEN(3)=0 THEN GOTO 3070
3090 IF PEN(8)>21 AND PEN(9)<23 THEN GOTO 3290
3100 LOCATE 23,1:PRINT "ANOTHER DESIGN?     YES    NO      "
3110 LINE (150,185)-(175,170),1,B:LINE (195,185)-(220,170),1,B
3120 PEN ON
3130 IF PEN(3)=0 THEN GOTO 3120
3140 IF PEN(8)>21 AND PEN(9)>24 THEN GOTO 3210
3150 LOCATE 23,1:PRINT "UPDATE DATA?        YES    NO      "
3160 LINE (150,185)-(175,170),1,B:LINE (195,185)-(220,170),1,B
3170 PEN ON
3180 IF PEN(3)=0 THEN GOTO 3170
3190 IF PEN(8)>21 AND PEN(9)>24 THEN GOTO 3280
3200 GOTO 3270
3210 LOCATE 23,1:PRINT "START AGAIN?        YES    NO      "
3220 LINE (150,185)-(175,170),1,B:LINE (195,185)-(220,170),1,B
3230 PEN ON
3240 IF PEN(3)=0 THEN GOTO 3230
3250 IF PEN(8)>21 AND PEN(9)<23 THEN GOTO 3260 ELSE STOP
3260 CHAIN MERGE "CREATE",10,DELETE 10-3250
3270 CHAIN MERGE "SPACE",10,ALL,DELETE 10-3260
3280 CHAIN MERGE "GEOMTRY",10,ALL,DELETE 10-3270
3290 CHAIN MERGE "SITE",10,ALL,DELETE 10-3280
3300 CHAIN MERGE "PLNFILE",10,ALL,DELETE 10-3290
```

```
10 '*************************************************************************
20 'SITE Subroutine
30 '*************************************************************************
40 'This routine allows the user to develop a site plan, place trees
50 'and easement setbacks on it and then orient a building layout on
60 'on the site.  The site and building plan may then be stored.
70 '
80 DIM TSPINE(4,2),TYARDPTS(4,2),TPINPTS(4,2),THALLPTS(12,2),TPLANPTS(20,10,4)
90 CLS:PEN OFF:SAVEFLAG=0
100 LOCATE 5,1:PRINT SPC(11) "***************"
110 LOCATE 6,1:PRINT SPC(11) "* SITE Routine *"
120 LOCATE 7,1:PRINT SPC(11) "***************"
130 FOR A=1 TO 500:NEXT A
140 CLS
150 LOCATE 1,13:PRINT "SITE Subroutine"
160 IF ELEVATIONFLAG=1 THEN ELEVATIONFLAG=0:DATANM$="B:ELVTEMP":GOTO 280
      ELSE GOTO 170
170 LOCATE 23,1:PRINT "NEW SITE?           YES   NO     "
180 LINE (150,185)-(175,170),1,B:LINE (195,185)-(220,170),1,B
190 PEN ON
200 IF PEN(3)=0 THEN GOTO 190
210 IF PEN(8)>21 AND PEN(9)<23 THEN GOTO 630
220 LINE (150,185)-(175,170),0,B:LINE (195,185)-(220,170),0,B
230 LOCATE 23,1:PRINT SPC(39)
240 LOCATE 3,1 :PRINT "PLANS ON DISK"
250 LOCATE 4,1
260 FILES "B:"
270 LOCATE 23,1:INPUT "ENTER FILE NAME OF SITE: ",DATANM$
280 OPEN DATANM$ FOR INPUT AS #2
290 INPUT#2,NUMSITEPOINTS
300 INPUT#2,NUMEASEPOINTS
310 INPUT#2,NUMTREES
320 FOR A=1 TO NUMSITEPOINTS
330     INPUT#2,SITE(A,1)
340     INPUT#2,SITE(A,2)
350 NEXT A
360 FOR A=1 TO NUMEASEPOINTS
370     INPUT#2,EASE(A,1)
380     INPUT#2,EASE(A,2)
390 NEXT A
400 FOR A=1 TO NUMTREES
410     INPUT#2,TREE(A,1)
420     INPUT#2,TREE(A,2)
430     INPUT#2,TREE(A,3)
440 NEXT A
450 CLOSE#2
460 CLS
470 FOR A=1 TO NUMSITEPOINTS-1
480     LINE (SITE(A,1),SITE(A,2))-(SITE(A+1,1),SITE(A+1,2)),1
490 NEXT A
500 LINE (SITE(A,1),SITE(A,2))-(SITE(1,1),SITE(1,2)),1
510 IF NUMEASEPOINTS=0 THEN GOTO 570
520 FOR A=1 TO NUMEASEPOINTS-1
530     LINE (EASE(A,1),EASE(A,2))-(EASE(A+1,1),EASE(A+1,2)),1
540 NEXT A
550 LINE (EASE(A,1),EASE(A,2))-(EASE(1,1),EASE(1,2)),1
560 PAINT (1,1),3,1
570 IF NUMTREES=0 THEN GOTO 620
580 FOR A=1 TO NUMTREES
590     CIRCLE(TREE(A,1),TREE(A,2)),TREE(A,3),1
```

```
600     PAINT (TREE(A,1),TREE(A,2)),3,1
610 NEXT A
620 GOTO 2140
630 CLS
640 LOCATE 22,37:PRINT "N"
650 LINE(285,160)-(300,160),1:LINE(300,160)-(292,152),1:
     LINE(292,152)-(285,160),1
660 PAINT (292,155),3,1
670 FLAG=1:A=0:SCALE=1
680 LOCATE 3,1:PRINT "   CORNER POINTS"
690 WHILE FLAG
700      A=A+1
710      LOCATE 23,1:INPUT "ENTER SITE CORNER POINTS   ",SITE(A,1),SITE(A,2)
720      LOCATE 23,26:PRINT SPC(12)
730      LOCATE A+4,1:PRINT SITE(A,1),SITE(A,2)
740      IF A=1 THEN GOTO 760
750      IF SITE(A,1)=SITE(1,1) AND SITE(A,2)=SITE(1,2) THEN FLAG=0
760 WEND
770 NUMSITEPOINTS=A-1
780 LOCATE 23,1:PRINT "ANY CHANGES?       YES   NO      "
790 LINE (150,185)-(175,170),1,B:LINE (195,185)-(220,170),1,B
800 PEN ON
810 IF PEN(3)=0 THEN GOTO 800
820 IF PEN(8)>21 AND PEN(9)>24 THEN GOTO 890
830 LOCATE 3,1:PRINT SPC(22)
840 FOR B=1 TO A
850     LOCATE B+4,1:PRINT SPC(39)
860 NEXT B
870 LINE (150,185)-(175,170),0,B:LINE (195,185)-(220,170),0,B
880 GOTO 670
890 LOCATE 3,1:PRINT SPC(39)
900 FOR B=1 TO A
910     LOCATE B+4,1:PRINT SPC(39)
920 NEXT B
930 LINE (150,185)-(175,170),0,B:LINE (195,185)-(220,170),0,B
940 FOR B=1 TO A-1
950     IF ABS(SITE(B+1,1)-SITE(B,1))>320 THEN TEMPSCALE1=310/(ABS(SITE(B+1,1)-
         SITE(B,1))+.1) ELSE TEMPSCALE1=1
960     IF ABS(SITE(B+1,2)-SITE(B,2))>140 THEN TEMPSCALE2=140/(ABS(SITE(B+1,2)-
         SITE(B,2))+.1) ELSE TEMPSCALE2=1
970     IF TEMPSCALE1<SCALE THEN SCALE=TEMPSCALE1
980     IF TEMPSCALE2<TEMPSCALE1 AND TEMPSCALE2<SCALE THEN SCALE=TEMPSCALE2
990 NEXT B
1000 FOR B=1 TO A
1010     SITE(B,1)=SITE(B,1)*SCALE
1020     SITE(B,2)=SITE(B,2)*SCALE
1030 NEXT B
1040 LOCATE  1,1:PRINT SPC(39)
1050 LOCATE 23,1:PRINT SPC(39)
1060 FOR B=1 TO A-1
1070     LINE (SITE(B,1),SITE(B,2))-(SITE(B+1,1),SITE(B+1,2)),1
1080 NEXT B
1090 LINE (SITE(A,1),SITE(A,2))-(SITE(1,1),SITE(1,2)),1
1100 LOCATE 23,1:PRINT "EASEMENT SETBACK?  YES   NO      "
1110 LINE (150,185)-(175,170),1,B:LINE (195,185)-(220,170),1,B
1120 PEN ON
1130 IF PEN(3)=0 THEN GOTO 1120
1140 IF PEN(8)>21 AND PEN(9)>24 THEN GOTO 1550
1150 LOCATE 23,1:PRINT SPC(39)
1160 LINE (150,185)-(175,170),0,B:LINE (195,185)-(220,170),0,B
```

```
1170 FLAG=1:A=0
1180 LOCATE 3,2:PRINT " EASEMENT POINTS"
1190 WHILE FLAG
1200     A=A+1
1210     LOCATE 23,1:INPUT "ENTER EASEMENT POINTS  ",EASE(A,1),EASE(A,2)
1220     LOCATE 23,23:PRINT SPC(15)
1230     LOCATE A+4,2:PRINT EASE(A,1),EASE(A,2)
1240     IF A=1 THEN GOTO 1260
1250     IF EASE(A,1)=EASE(1,1) AND EASE(A,2)=EASE(1,2) THEN FLAG=0
1260 WEND
1270 NUMEASEPOINTS=A-1
1280 LOCATE 23,1:PRINT "ANY CHANGES?       YES   NO     "
1290 LINE (150,185)-(175,170),1,B:LINE (195,185)-(220,170),1,B
1300 PEN ON
1310 IF PEN(3)=0 THEN GOTO 1300
1320 IF PEN(8)>21 AND PEN(9)>24 THEN GOTO 1400
1330 LOCATE 3,2:PRINT SPC(22)
1340 FOR B=1 TO A
1350     LOCATE B+4,2:PRINT SPC(22)
1360 NEXT B
1370 LINE (150,185)-(175,170),0,B:LINE (195,185)-(220,170),0,B
1380 LOCATE 23,1:PRINT SPC(39)
1390 GOTO 1170
1400 LOCATE 3,2:PRINT SPC(22)
1410 FOR B=1 TO A
1420     LOCATE B+4,2:PRINT SPC(22)
1430 NEXT B
1440 LINE (150,185)-(175,170),0,B:LINE (195,185)-(220,170),0,B
1450 LOCATE 23,1:PRINT SPC(39)
1460 FOR B=1 TO A
1470     EASE(B,1)=EASE(B,1)*SCALE
1480     EASE(B,2)=EASE(B,2)*SCALE
1490 NEXT B
1500 FOR B=1 TO A-1
1510     LINE (EASE(B,1),EASE(B,2))-(EASE(B+1,1),EASE(B+1,2)),1
1520 NEXT B
1530 LINE (EASE(A,1),EASE(A,2))-(EASE(1,1),EASE(1,2)),1
1540 PAINT (1,1),3,1
1550 LOCATE 23,1:PRINT "TREE PLACEMENT?    YES   NO     "
1560 LINE (150,185)-(175,170),1,B:LINE (195,185)-(220,170),1,B
1570 PEN ON
1580 IF PEN(3)=0 THEN GOTO 1570
1590 IF PEN(8)>21 AND PEN(9)>24 THEN GOTO 2140
1600 LINE (150,185)-(175,170),0,B:LINE (195,185)-(220,170),0,B
1610 FLAG=1:A=0
1620  WHILE FLAG
1630    LOCATE 23,1:PRINT "INDICATE TREE LOCATIONS          "
1640    A=A+1
1650    FOR B=1 TO 500:NEXT B
1660    PEN ON
1670    IF PEN(3)=0 THEN GOTO 1660
1680    X=PEN(4):Y=PEN(5)
1690    PSET (X,Y)
1700    LOCATE 23,1:PRINT "CORRECT POINT?     YES   NO     "
1710    LINE (150,185)-(175,170),1,B:LINE (195,185)-(220,170),1,B
1720    PEN ON
1730    IF PEN(3)=0 THEN GOTO 1720
1740    IF PEN(8)>21 AND PEN(9)>24 THEN PSET (X,Y),0:GOTO 1650 ELSE GOTO 1750
1750    PSET (X,Y),0
1760    TREE(A,1)=X:TREE(A,2)=Y
```

```
1770    LOCATE 23,1:PRINT SPC(39)
1780    LINE (150,185)-(175,170),0,B:LINE (195,185)-(220,170),0,B
1790    LOCATE 23,1:INPUT "ENTER TREE SIZE  ",TREE(A,3)
1800    TREE(A,3)=(TREE(A,3)/2)*SCALE
1810    CIRCLE (X,Y),TREE(A,3),1
1820    PAINT (X,Y),3,1
1830    LOCATE 23,1:PRINT "ANOTHER TREE?      YES   NO     "
1840    LINE (150,185)-(175,170),1,B:LINE (195,185)-(220,170),1,B
1850    PEN ON
1860    IF PEN(3)=0 THEN GOTO 1850
1870    IF PEN(8)>21 AND PEN(9)>24 THEN FLAG=0
1880    LINE (150,185)-(175,170),0,B:LINE (195,185)-(220,170),0,B
1890 WEND
1900 NUMTREES=A
1910 LOCATE 23,1:PRINT "SAVE SITE?          YES   NO     "
1920 LINE (150,185)-(175,170),1,B:LINE (195,185)-(220,170),1,B
1930 PEN ON
1940 IF PEN(3)=0 THEN GOTO 1930
1950 IF PEN(8)>21 AND PEN(9)<23 THEN JUMP$="SITE1":GOTO 6680 ELSE GOTO 1960
1960 LINE (150,185)-(175,170),0,B:LINE (195,185)-(220,170),0,B
1970 LOCATE 1,1:PRINT SPC(39):LOCATE 23,1:PRINT SPC(39)
1980 FOR A=1 TO NUMSITEPOINTS-1
1990     LINE (SITE(A,1),SITE(A,2))-(SITE(A+1,1),SITE(A+1,2)),1
2000 NEXT A
2010 LINE (SITE(A,1),SITE(A,2))-(SITE(1,1),SITE(1,2)),1
2020 IF NUMEASEPOINTS=0 THEN GOTO 2080
2030 FOR A=1 TO NUMEASEPOINTS-1
2040     LINE (EASE(A,1),EASE(A,2))-(EASE(A+1,1),EASE(A+1,2)),1
2050 NEXT A
2060 LINE (EASE(A,1),EASE(A,2))-(EASE(1,1),EASE(1,2)),1
2070 PAINT (1,1),3,1
2080 IF NUMTREES=0 THEN GOTO 2130
2090 FOR A=1 TO NUMTREES
2100     CIRCLE(TREE(A,1),TREE(A,2)),TREE(A,3),1
2110     PAINT (TREE(A,1),TREE(A,2)),3,1
2120 NEXT A
2130 '
2140 LOCATE 23,1:PRINT "USE CURRENT PLAN?  YES   NO     "
2150 LINE (150,185)-(175,170),1,B:LINE (195,185)-(220,170),1,B
2160 PEN ON
2170 IF PEN(3)=0 THEN GOTO 2160
2180 IF PEN(8)>21 AND PEN(9)<23 THEN GOTO 2740
2190 LINE (150,185)-(175,170),0,B:LINE (195,185)-(220,170),0,B
2200 LOCATE 23,1:PRINT SPC(39)
2210 'LOCATE 18,1:PRINT "PLANS ON DISK"
2220 LOCATE 18,1
2230 FILES "B:"
2240 LOCATE 23,1:INPUT "ENTER FILE NAME FOR PLAN: ",DATANM2$
2250 LOCATE 18,1:PRINT SPC(35)
2260 LOCATE 19,1:PRINT SPC(35)
2270 LOCATE 20,1:PRINT SPC(35)
2280 LOCATE 21,1:PRINT SPC(39)
2290 LOCATE 22,1:PRINT SPC(30)
2300 OPEN DATANM2$ FOR INPUT AS #1
2310 INPUT#1,JUMP
2320 INPUT#1,N
2330 IF JUMP=1 THEN GOTO 2370
2340 IF JUMP=2 THEN GOTO 2420
2350 IF JUMP=3 THEN GOTO 2470
2360 IF JUMP=4 THEN GOTO 2550
```

```
2370 FOR A=1 TO 4
2380     INPUT#1,SPINE(A,1)
2390     INPUT#1,SPINE(A,2)
2400 NEXT A
2410 GOTO 2550
2420 FOR A=1 TO 4
2430     INPUT#1,YARDPTS(A,1)
2440     INPUT#1,YARDPTS(A,2)
2450 NEXT A
2460 GOTO 2550
2470 FOR A=1 TO 4
2480    INPUT#1,PINPTS(A,1)
2490    INPUT#1,PINPTS(A,2)
2500 NEXT A
2510 FOR A=1 TO 12
2520    INPUT#1,HALLPTS(A,1)
2530    INPUT#1,HALLPTS(A,2)
2540 NEXT A
2550 FOR A=1 TO N
2560    INPUT#1,SPACE(A,4)
2570    INPUT#1,SPACE(A,11)
2580 NEXT A
2590 FOR A=1 TO N
2600    INPUT#1,TRACK(A,1)
2610 NEXT A
2620 FOR A=1 TO N
2630    FOR B=1 TO SPACE(A,11)
2640       INPUT#1,PLANPTS(A,B,1)
2650       INPUT#1,PLANPTS(A,B,2)
2660       INPUT#1,PLANPTS(A,B,3)
2670       INPUT#1,PLANPTS(A,B,4)
2680    NEXT B
2690 NEXT A
2700 FOR A=1 TO N
2710    INPUT#1,TITLE$(A,A)
2720 NEXT A
2730 CLOSE#1
2740 LOCATE 23,1:PRINT SPC(39)
2750 FOR A=1 TO N
2760    FOR B=1 TO SPACE(A,11)
2770       PLANPTS(A,B,2)=PLANPTS(A,B,2)*SCALE
2780       PLANPTS(A,B,3)=PLANPTS(A,B,3)*SCALE
2790    NEXT B
2800 NEXT A
2810 IF JUMP=1 THEN GOTO 2850
2820 IF JUMP=2 THEN GOTO 2900
2830 IF JUMP=3 THEN GOTO 2950
2840 IF JUMP=4 THEN GOTO 3240
2850 SPINE(1,1)=SPINE(1,1)*SCALE
2860 SPINE(1,2)=SPINE(1,2)*SCALE
2870 SPINE(3,1)=SPINE(3,1)*SCALE
2880 SPINE(3,2)=SPINE(3,2)*SCALE
2890 GOTO 3020
2900 YARDPTS(1,1)=YARDPTS(1,1)*SCALE
2910 YARDPTS(1,2)=YARDPTS(1,2)*SCALE
2920 YARDPTS(3,1)=YARDPTS(3,1)*SCALE
2930 YARDPTS(3,2)=YARDPTS(3,2)*SCALE
2940 GOTO 3070
2950 FOR A=1 TO 4
2960    PINPTS(A,1)=PINPTS(A,1)*SCALE
```

```
2970     PINPTS(A,2)=PINPTS(A,2)*SCALE
2980     HALLPTS(A,1)=HALLPTS(A,1)*SCALE
2990     HALLPTS(A,2)=HALLPTS(A,2)*SCALE
3000 NEXT A
3010 GOTO 3120
3020 '
3030 'Draw spine and activities
3040 '
3050 LINE (SPINE(1,1),SPINE(1,2))-(SPINE(3,1),SPINE(3,2)),1,B
3060 GOTO 3240
3070 '
3080 'Draw courtyard and activities
3090 '
3100 LINE (YARDPTS(1,1),YARDPTS(1,2))-(YARDPTS(3,1),YARDPTS(3,2)),1,B
3110 GOTO 3240
3120 '
3130 'Draw pinwheel and activities
3140 '
3150 LINE (PINPTS(1,1),PINPTS(1,2))-(PINPTS(3,1),PINPTS(3,2)),1,B
3160 LINE (HALLPTS(1,1),HALLPTS(1,2))-(PINPTS(4,1),PINPTS(4,2)),1,B
3170 LINE (HALLPTS(2,1),HALLPTS(2,2))-(PINPTS(1,1),PINPTS(1,2)),1,B
3180 LINE (HALLPTS(3,1),HALLPTS(3,2))-(PINPTS(2,1),PINPTS(2,2)),1,B
3190 LINE (HALLPTS(4,1),HALLPTS(4,2))-(PINPTS(3,1),PINPTS(3,2)),1,B
3200 GOTO 3240
3210 '
3220 'Draw activities
3230 '
3240 FOR A=1 TO N
3250 FLAG=1:I=1:COUNT=0
3260 WHILE FLAG
3270     I=PLANPTS(TRACK(A,1),I,1)
3280     LINE (PLANPTS(TRACK(A,1),PLANPTS(TRACK(A,1),I,1),2),PLANPTS(TRACK(A,1),
          PLANPTS(TRACK(A,1),I,1),3))-(PLANPTS(TRACK(A,1),PLANPTS(TRACK(A,1),I,4
          ),2),PLANPTS(TRACK(A,1),PLANPTS(TRACK(A,1),I,4),3)),1
3290     I=PLANPTS(TRACK(A,1),I,4)
3300     COUNT=COUNT+1
3310     IF COUNT>=SPACE(TRACK(A,1),11) THEN FLAG=0
3320 WEND
3330 NEXT A
3340 LOCATE 23,1:PRINT " RELOCATE   ROTATE   COMPLETE      "
3350 LINE (2,185)-(75,170),1,B:LINE (82,185)-(150,170),1,B:
      LINE(160,185)-(240,170),1,B
3360 PEN ON
3370 IF PEN(3)=0 THEN GOTO 3360
3380 IF PEN(8)>21 AND PEN(9)<10 THEN GOTO 3410
3390 IF PEN(8)>21 AND PEN(9)>10 AND PEN(9)<20 THEN GOTO 4470
3400 IF PEN(8)>21 AND PEN(9)>20 THEN GOTO 6250
3410 '
3420 'Translation of building
3430 '
3440 LINE (2,185)-(75,170),0,B:LINE (82,185)-(150,170),0,B:
      LINE(160,185)-(240,170),0,B
3450 LOCATE 23,1:PRINT SPC(39)
3460 LOCATE 23,1:PRINT "INDICATE NEW LOCATION           "
3470 FOR A=1 TO 500:NEXT A
3480 PEN ON
3490 IF PEN(3)=0 THEN GOTO 3480
3500 X=PEN(4):Y=PEN(5)
3510 PSET (X,Y)
3520 LOCATE 23,1:PRINT "CORRECT POINT?     YES   NO      "
```

```
3530 LINE (150,185)-(175,170),1,B:LINE (195,185)-(220,170),1,B
3540 PEN ON
3550 IF PEN(3)=0 THEN GOTO 3540
3560 IF PEN(8)>21 AND PEN(9)>24 THEN PSET (X,Y),0:GOTO 3470 ELSE GOTO 3570
3570 PSET (X,Y),0
3580 LINE (150,185)-(175,170),0,B:LINE (195,185)-(220,170),0,B
3590 LOCATE 23,1:PRINT SPC(39)
3600 IF JUMP=1 THEN GOTO 3640
3610 IF JUMP=2 THEN GOTO 3780
3620 IF JUMP=3 THEN GOTO 3920
3630 IF JUMP=4 THEN GOTO 4260 ELSE STOP
3640 XDIST=X-SPINE(1,1):YDIST=Y-SPINE(1,2)
3650 LINE (SPINE(1,1),SPINE(1,2))-(SPINE(2,1),SPINE(2,2)),0
3660 LINE (SPINE(2,1),SPINE(2,2))-(SPINE(3,1),SPINE(3,2)),0
3670 LINE (SPINE(3,1),SPINE(3,2))-(SPINE(4,1),SPINE(4,2)),0
3680 LINE (SPINE(4,1),SPINE(4,2))-(SPINE(1,1),SPINE(1,2)),0
3690 FOR A=1 TO 4
3700    SPINE(A,1)=SPINE(A,1)+XDIST
3710    SPINE(A,2)=SPINE(A,2)+YDIST
3720 NEXT A
3730 LINE (SPINE(1,1),SPINE(1,2))-(SPINE(2,1),SPINE(2,2)),1
3740 LINE (SPINE(2,1),SPINE(2,2))-(SPINE(3,1),SPINE(3,2)),1
3750 LINE (SPINE(3,1),SPINE(3,2))-(SPINE(4,1),SPINE(4,2)),1
3760 LINE (SPINE(4,1),SPINE(4,2))-(SPINE(1,1),SPINE(1,2)),1
3770 GOTO 4270
3780 XDIST=X-YARDPTS(1,1):YDIST=Y-YARDPTS(1,2)
3790 LINE (YARDPTS(1,1),YARDPTS(1,2))-(YARDPTS(2,1),YARDPTS(2,2)),0
3800 LINE (YARDPTS(2,1),YARDPTS(2,2))-(YARDPTS(3,1),YARDPTS(3,2)),0
3810 LINE (YARDPTS(3,1),YARDPTS(3,2))-(YARDPTS(4,1),YARDPTS(4,2)),0
3820 LINE (YARDPTS(4,1),YARDPTS(4,2))-(YARDPTS(1,1),YARDPTS(1,2)),0
3830 FOR A=1 TO 4
3840    YARDPTS(A,1)=YARDPTS(A,1)+XDIST
3850    YARDPTS(A,2)=YARDPTS(A,2)+YDIST
3860 NEXT A
3870 LINE (YARDPTS(1,1),YARDPTS(1,2))-(YARDPTS(2,1),YARDPTS(2,2)),1
3880 LINE (YARDPTS(2,1),YARDPTS(2,2))-(YARDPTS(3,1),YARDPTS(3,2)),1
3890 LINE (YARDPTS(3,1),YARDPTS(3,2))-(YARDPTS(4,1),YARDPTS(4,2)),1
3900 LINE (YARDPTS(4,1),YARDPTS(4,2))-(YARDPTS(1,1),YARDPTS(1,2)),1
3910 GOTO 4270
3920 XDIST=X-PINPTS(1,1):YDIST=Y-PINPTS(1,2)
3930 LINE (HALLPTS(12,1),HALLPTS(12,2))-(HALLPTS(1,1),HALLPTS(1,2)),0
3940 LINE (HALLPTS(1,1),HALLPTS(1,2))-(HALLPTS(5,1),HALLPTS(5,2)),0
3950 LINE (HALLPTS(5,1),HALLPTS(5,2))-(PINPTS(4,1),PINPTS(4,2)),0
3960 LINE (HALLPTS(9,1),HALLPTS(9,2))-(HALLPTS(2,1),HALLPTS(2,2)),0
3970 LINE (HALLPTS(2,1),HALLPTS(2,2))-(HALLPTS(6,1),HALLPTS(6,2)),0
3980 LINE (HALLPTS(6,1),HALLPTS(6,2))-(PINPTS(1,1),PINPTS(1,2)),0
3990 LINE (PINPTS(2,1),PINPTS(2,2))-(HALLPTS(7,1),HALLPTS(7,2)),0
4000 LINE (HALLPTS(7,1),HALLPTS(7,2))-(HALLPTS(3,1),HALLPTS(3,2)),0
4010 LINE (HALLPTS(3,1),HALLPTS(3,2))-(HALLPTS(10,1),HALLPTS(10,2)),0
4020 LINE (PINPTS(3,1),PINPTS(3,2))-(HALLPTS(8,1),HALLPTS(8,2)),0
4030 LINE (HALLPTS(8,1),HALLPTS(8,2))-(HALLPTS(4,1),HALLPTS(4,2)),0
4040 LINE (HALLPTS(4,1),HALLPTS(4,2))-(HALLPTS(11,1),HALLPTS(11,2)),0
4050 FOR A=1 TO 4
4060    PINPTS(A,1)=PINPTS(A,1)+XDIST
4070    PINPTS(A,2)=PINPTS(A,2)+YDIST
4080 NEXT A
4090 FOR A=1 TO 12
4100    HALLPTS(A,1)=HALLPTS(A,1)+XDIST
4110    HALLPTS(A,2)=HALLPTS(A,2)+YDIST
4120 NEXT A
```

```
4130 LINE (HALLPTS(12,1),HALLPTS(12,2))-(HALLPTS(1,1),HALLPTS(1,2)),1
4140 LINE (HALLPTS(1,1),HALLPTS(1,2))-(HALLPTS(5,1),HALLPTS(5,2)),1
4150 LINE (HALLPTS(5,1),HALLPTS(5,2))-(PINPTS(4,1),PINPTS(4,2)),1
4160 LINE (HALLPTS(9,1),HALLPTS(9,2))-(HALLPTS(2,1),HALLPTS(2,2)),1
4170 LINE (HALLPTS(2,1),HALLPTS(2,2))-(HALLPTS(6,1),HALLPTS(6,2)),1
4180 LINE (HALLPTS(6,1),HALLPTS(6,2))-(PINPTS(1,1),PINPTS(1,2)),1
4190 LINE (PINPTS(2,1),PINPTS(2,2))-(HALLPTS(7,1),HALLPTS(7,2)),1
4200 LINE (HALLPTS(7,1),HALLPTS(7,2))-(HALLPTS(3,1),HALLPTS(3,2)),1
4210 LINE (HALLPTS(3,1),HALLPTS(3,2))-(HALLPTS(10,1),HALLPTS(10,2)),1
4220 LINE (PINPTS(3,1),PINPTS(3,2))-(HALLPTS(8,1),HALLPTS(8,2)),1
4230 LINE (HALLPTS(8,1),HALLPTS(8,2))-(HALLPTS(4,1),HALLPTS(4,2)),1
4240 LINE (HALLPTS(4,1),HALLPTS(4,2))-(HALLPTS(11,1),HALLPTS(11,2)),1
4250 GOTO 4270
4260 XDIST=X-PLANPTS(1,1,2):YDIST=Y-PLANPTS(1,1,3)
4270 '
4280 'Translate to new coordinates
4290 '
4300 FOR A=1 TO N
4310 FLAG=1:I=1:COUNT=0
4320 WHILE FLAG
4330    I=PLANPTS(TRACK(A,1),I,1)
4340    LINE (PLANPTS(TRACK(A,1),PLANPTS(TRACK(A,1),I,1),2),PLANPTS(TRACK(A,1),
         PLANPTS(TRACK(A,1),I,1),3))-(PLANPTS(TRACK(A,1),PLANPTS(TRACK(A,1),I,4
         ),2),PLANPTS(TRACK(A,1),PLANPTS(TRACK(A,1),I,4),3)),0
4350    I=PLANPTS(TRACK(A,1),I,4)
4360    COUNT=COUNT+1
4370    IF COUNT>=SPACE(TRACK(A,1),11) THEN FLAG=0
4380 WEND
4390 NEXT A
4400 FOR A=1 TO N
4410    FOR B=1 TO SPACE(A,11)
4420      PLANPTS(A,B,2)=PLANPTS(A,B,2)+XDIST
4430      PLANPTS(A,B,3)=PLANPTS(A,B,3)+YDIST
4440    NEXT B
4450 NEXT A
4460 GOTO 3210
4470 '
4480 'Rotation of building
4490 '
4500 FOR A=1 TO N
4510    FOR B=1 TO SPACE(A,11)
4520      TPLANPTS(A,B,2)=PLANPTS(A,B,2)
4530      TPLANPTS(A,B,3)=PLANPTS(A,B,3)
4540    NEXT B
4550 NEXT A
4560 LINE (2,185)-(75,170),0,B:LINE (82,185)-(150,170),0,B:
     LINE(160,185)-(240,170),0,B
4570 LOCATE 23,1:PRINT SPC(39)
4580 LOCATE 23,1:INPUT "ENTER DEGREES ROTATION  ",DEGREE
4590 DEGREE=DEGREE*(-3.141593/180)
4600 IF JUMP=1 THEN GOTO 4640
4610 IF JUMP=2 THEN GOTO 5030
4620 IF JUMP=3 THEN GOTO 5420
4630 IF JUMP=4 THEN GOTO 6050
4640 '
4650 'Rotate spine plan around spine corner
4660 '
4670 LINE (SPINE(1,1),SPINE(1,2))-(SPINE(2,1),SPINE(2,2)),0
4680 LINE (SPINE(2,1),SPINE(2,2))-(SPINE(3,1),SPINE(3,2)),0
4690 LINE (SPINE(3,1),SPINE(3,2))-(SPINE(4,1),SPINE(4,2)),0
```

```
4700 LINE (SPINE(4,1),SPINE(4,2))-(SPINE(1,1),SPINE(1,2)),0
4710 FOR A=2 TO 4
4720    TSPINE(A,1)=SPINE(A,1)
4730    TSPINE(A,2)=SPINE(A,2)
4740 NEXT A
4750 FOR A=2 TO 4
4760     SPINE(A,1)=TSPINE(A,1)*COS(DEGREE)-TSPINE(A,2)*SIN(DEGREE)+
            SPINE(1,1)*(1-COS(DEGREE))+SPINE(1,2)*SIN(DEGREE)
4770     SPINE(A,2)=TSPINE(A,1)*SIN(DEGREE)+TSPINE(A,2)*COS(DEGREE)+
            SPINE(1,2)*(1-COS(DEGREE))-SPINE(1,1)*SIN(DEGREE)
4780 NEXT A
4790 FOR A=1 TO N
4800 FLAG=1:I=1:COUNT=0
4810 WHILE FLAG
4820    I=PLANPTS(TRACK(A,1),I,1)
4830    LINE (PLANPTS(TRACK(A,1),PLANPTS(TRACK(A,1),I,1),2),PLANPTS(TRACK(A,1),
         PLANPTS(TRACK(A,1),I,1),3))-(PLANPTS(TRACK(A,1),PLANPTS(TRACK(A,1),I,4
         ),2),PLANPTS(TRACK(A,1),PLANPTS(TRACK(A,1),I,4),3)),0
4840    I=PLANPTS(TRACK(A,1),I,4)
4850    COUNT=COUNT+1
4860    IF COUNT>=SPACE(TRACK(A,1),11) THEN FLAG=0
4870 WEND
4880 NEXT A
4890 FOR A=1 TO N
4900    FOR B=1 TO SPACE(A,11)
4910     PLANPTS(A,B,2)=TPLANPTS(A,B,2)*COS(DEGREE)-TPLANPTS(A,B,3)*SIN(DEGREE)+
            SPINE(1,1)*(1-COS(DEGREE))+SPINE(1,2)*SIN(DEGREE)
4920     PLANPTS(A,B,3)=TPLANPTS(A,B,2)*SIN(DEGREE)+TPLANPTS(A,B,3)*COS(DEGREE)+
            SPINE(1,2)*(1-COS(DEGREE))-SPINE(1,1)*SIN(DEGREE)
4930    NEXT B
4940 NEXT A
4950 '
4960 'Draw rotated spine
4970 '
4980 LINE (SPINE(1,1),SPINE(1,2))-(SPINE(2,1),SPINE(2,2)),1
4990 LINE (SPINE(2,1),SPINE(2,2))-(SPINE(3,1),SPINE(3,2)),1
5000 LINE (SPINE(3,1),SPINE(3,2))-(SPINE(4,1),SPINE(4,2)),1
5010 LINE (SPINE(4,1),SPINE(4,2))-(SPINE(1,1),SPINE(1,2)),1
5020 GOTO 3210
5030 '
5040 'Rotate courtyard and activities
5050 '
5060 LINE (YARDPTS(1,1),YARDPTS(1,2))-(YARDPTS(2,1),YARDPTS(2,2)),0
5070 LINE (YARDPTS(2,1),YARDPTS(2,2))-(YARDPTS(3,1),YARDPTS(3,2)),0
5080 LINE (YARDPTS(3,1),YARDPTS(3,2))-(YARDPTS(4,1),YARDPTS(4,2)),0
5090 LINE (YARDPTS(4,1),YARDPTS(4,2))-(YARDPTS(1,1),YARDPTS(1,2)),0
5100 FOR A=2 TO 4
5110    TYARDPTS(A,1)=YARDPTS(A,1)
5120    TYARDPTS(A,2)=YARDPTS(A,2)
5130 NEXT A
5140 FOR A=2 TO 4
5150    YARDPTS(A,1)=TYARDPTS(A,1)*COS(DEGREE)-TYARDPTS(A,2)*SIN(DEGREE)+
           YARDPTS(1,1)*(1-COS(DEGREE))+YARDPTS(1,2)*SIN(DEGREE)
5160    YARDPTS(A,2)=TYARDPTS(A,1)*SIN(DEGREE)+TYARDPTS(A,2)*COS(DEGREE)+
           YARDPTS(1,2)*(1-COS(DEGREE))-YARDPTS(1,1)*SIN(DEGREE)
5170 NEXT A
5180 FOR A=1 TO N
5190 FLAG=1:I=1:COUNT=0
5200 WHILE FLAG
5210    I=PLANPTS(TRACK(A,1),I,1)
```

```
5220     LINE (PLANPTS(TRACK(A,1),PLANPTS(TRACK(A,1),I,1),2),PLANPTS(TRACK(A,1),
         PLANPTS(TRACK(A,1),I,1),3))-(PLANPTS(TRACK(A,1),PLANPTS(TRACK(A,1),I,4
         ),2),PLANPTS(TRACK(A,1),PLANPTS(TRACK(A,1),I,4),3)),0
5230     I=PLANPTS(TRACK(A,1),I,4)
5240     COUNT=COUNT+1
5250     IF COUNT>=SPACE(TRACK(A,1),11) THEN FLAG=0
5260 WEND
5270 NEXT A
5280 FOR A=1 TO N
5290     FOR B=1 TO SPACE(A,11)
5300       PLANPTS(A,B,2)=TPLANPTS(A,B,2)*COS(DEGREE)-TPLANPTS(A,B,3)*SIN(DEGREE)+
             YARDPTS(1,1)*(1-COS(DEGREE))+YARDPTS(1,2)*SIN(DEGREE)
5310       PLANPTS(A,B,3)=TPLANPTS(A,B,2)*SIN(DEGREE)+TPLANPTS(A,B,3)*COS(DEGREE)+
             YARDPTS(1,2)*(1-COS(DEGREE))-YARDPTS(1,1)*SIN(DEGREE)
5320     NEXT B
5330 NEXT A
5340 '
5350 'Draw rotated courtyard
5360 '
5370 LINE (YARDPTS(1,1),YARDPTS(1,2))-(YARDPTS(2,1),YARDPTS(2,2)),1
5380 LINE (YARDPTS(2,1),YARDPTS(2,2))-(YARDPTS(3,1),YARDPTS(3,2)),1
5390 LINE (YARDPTS(3,1),YARDPTS(3,2))-(YARDPTS(4,1),YARDPTS(4,2)),1
5400 LINE (YARDPTS(4,1),YARDPTS(4,2))-(YARDPTS(1,1),YARDPTS(1,2)),1
5410 GOTO 3210
5420 '
5430 'Rotate pinwheel and activities
5440 '
5450 LINE (HALLPTS(12,1),HALLPTS(12,2))-(HALLPTS(1,1),HALLPTS(1,2)),0
5460 LINE (HALLPTS(1,1),HALLPTS(1,2))-(HALLPTS(5,1),HALLPTS(5,2)),0
5470 LINE (HALLPTS(5,1),HALLPTS(5,2))-(PINPTS(4,1),PINPTS(4,2)),0
5480 LINE (HALLPTS(9,1),HALLPTS(9,2))-(HALLPTS(2,1),HALLPTS(2,2)),0
5490 LINE (HALLPTS(2,1),HALLPTS(2,2))-(HALLPTS(6,1),HALLPTS(6,2)),0
5500 LINE (HALLPTS(6,1),HALLPTS(6,2))-(PINPTS(1,1),PINPTS(1,2)),0
5510 LINE (PINPTS(2,1),PINPTS(2,2))-(HALLPTS(7,1),HALLPTS(7,2)),0
5520 LINE (HALLPTS(7,1),HALLPTS(7,2))-(HALLPTS(3,1),HALLPTS(3,2)),0
5530 LINE (HALLPTS(3,1),HALLPTS(3,2))-(HALLPTS(10,1),HALLPTS(10,2)),0
5540 LINE (PINPTS(3,1),PINPTS(3,2))-(HALLPTS(8,1),HALLPTS(8,2)),0
5550 LINE (HALLPTS(8,1),HALLPTS(8,2))-(HALLPTS(4,1),HALLPTS(4,2)),0
5560 LINE (HALLPTS(4,1),HALLPTS(4,2))-(HALLPTS(11,1),HALLPTS(11,2)),0
5570 FOR A=2 TO 4
5580     TPINPTS(A,1)=PINPTS(A,1)
5590     TPINPTS(A,2)=PINPTS(A,2)
5600 NEXT A
5610 FOR A=2 TO 4
5620     PINPTS(A,1)=TPINPTS(A,1)*COS(DEGREE)-TPINPTS(A,2)*SIN(DEGREE)+
           PINPTS(1,1)*(1-COS(DEGREE))+PINPTS(1,2)*SIN(DEGREE)
5630     PINPTS(A,2)=TPINPTS(A,1)*SIN(DEGREE)+TPINPTS(A,2)*COS(DEGREE)+
           PINPTS(1,2)*(1-COS(DEGREE))-PINPTS(1,1)*SIN(DEGREE)
5640 NEXT A
5650 FOR A=1 TO 12
5660     THALLPTS(A,1)=HALLPTS(A,1)
5670     THALLPTS(A,2)=HALLPTS(A,2)
5680 NEXT A
5690 FOR A=1 TO 12
5700     HALLPTS(A,1)=THALLPTS(A,1)*COS(DEGREE)-THALLPTS(A,2)*SIN(DEGREE)+
           PINPTS(1,1)*(1-COS(DEGREE))+PINPTS(1,2)*SIN(DEGREE)
5710     HALLPTS(A,2)=THALLPTS(A,1)*SIN(DEGREE)+THALLPTS(A,2)*COS(DEGREE)+
           PINPTS(1,2)*(1-COS(DEGREE))-PINPTS(1,1)*SIN(DEGREE)
5720 NEXT A
5730 FOR A=1 TO N
```

```
5740 FLAG=1:I=1:COUNT=0
5750 WHILE FLAG
5760    I=PLANPTS(TRACK(A,1),I,1)
5770    LINE (PLANPTS(TRACK(A,1),PLANPTS(TRACK(A,1),I,1),2),PLANPTS(TRACK(A,1),
         PLANPTS(TRACK(A,1),I,1),3))-(PLANPTS(TRACK(A,1),PLANPTS(TRACK(A,1),I,4
         ),2),PLANPTS(TRACK(A,1),PLANPTS(TRACK(A,1),I,4),3)),0
5780    I=PLANPTS(TRACK(A,1),I,4)
5790    COUNT=COUNT+1
5800    IF COUNT>=SPACE(TRACK(A,1),11) THEN FLAG=0
5810 WEND
5820 NEXT A
5830 FOR A=1 TO N
5840    FOR B=1 TO SPACE(A,11)
5850     PLANPTS(A,B,2)=TPLANPTS(A,B,2)*COS(DEGREE)-TPLANPTS(A,B,3)*SIN(DEGREE)+
            PINPTS(1,1)*(1-COS(DEGREE))+PINPTS(1,2)*SIN(DEGREE)
5860     PLANPTS(A,B,3)=TPLANPTS(A,B,2)*SIN(DEGREE)+TPLANPTS(A,B,3)*COS(DEGREE)+
            PINPTS(1,2)*(1-COS(DEGREE))-PINPTS(1,1)*SIN(DEGREE)
5870    NEXT B
5880 NEXT A
5890 '
5900 'Draw rotated pinwheel
5910 '
5920 LINE (HALLPTS(12,1),HALLPTS(12,2))-(HALLPTS(1,1),HALLPTS(1,2)),1
5930 LINE (HALLPTS(1,1),HALLPTS(1,2))-(HALLPTS(5,1),HALLPTS(5,2)),1
5940 LINE (HALLPTS(5,1),HALLPTS(5,2))-(PINPTS(4,1),PINPTS(4,2)),1
5950 LINE (HALLPTS(9,1),HALLPTS(9,2))-(HALLPTS(2,1),HALLPTS(2,2)),1
5960 LINE (HALLPTS(2,1),HALLPTS(2,2))-(HALLPTS(6,1),HALLPTS(6,2)),1
5970 LINE (HALLPTS(6,1),HALLPTS(6,2))-(PINPTS(1,1),PINPTS(1,2)),1
5980 LINE (PINPTS(2,1),PINPTS(2,2))-(HALLPTS(7,1),HALLPTS(7,2)),1
5990 LINE (HALLPTS(7,1),HALLPTS(7,2))-(HALLPTS(3,1),HALLPTS(3,2)),1
6000 LINE (HALLPTS(3,1),HALLPTS(3,2))-(HALLPTS(10,1),HALLPTS(10,2)),1
6010 LINE (PINPTS(3,1),PINPTS(3,2))-(HALLPTS(8,1),HALLPTS(8,2)),1
6020 LINE (HALLPTS(8,1),HALLPTS(8,2))-(HALLPTS(4,1),HALLPTS(4,2)),1
6030 LINE (HALLPTS(4,1),HALLPTS(4,2))-(HALLPTS(11,1),HALLPTS(11,2)),1
6040 GOTO 3210
6050 '
6060 'Rotate clustered activities
6070 '
6080 FOR A=1 TO N
6090 FLAG=1:I=1:COUNT=0
6100 WHILE FLAG
6110    I=PLANPTS(TRACK(A,1),I,1)
6120    LINE (PLANPTS(TRACK(A,1),PLANPTS(TRACK(A,1),I,1),2),PLANPTS(TRACK(A,1),
         PLANPTS(TRACK(A,1),I,1),3))-(PLANPTS(TRACK(A,1),PLANPTS(TRACK(A,1),I,4
         ),2),PLANPTS(TRACK(A,1),PLANPTS(TRACK(A,1),I,4),3)),0
6130    I=PLANPTS(TRACK(A,1),I,4)
6140    COUNT=COUNT+1
6150    IF COUNT>=SPACE(TRACK(A,1),11) THEN FLAG=0
6160 WEND
6170 NEXT A
6180 FOR A=1 TO N
6190    FOR B=1 TO SPACE(A,11)
6200     PLANPTS(A,B,2)=TPLANPTS(A,B,2)*COS(DEGREE)-TPLANPTS(A,B,3)*SIN(DEGREE)+
            PLANPTS(TRACK(1,1),1,2)*(1-COS(DEGREE))
            +PLANPTS(TRACK(1,1),1,3)*SIN(DEGREE)
6210     PLANPTS(A,B,3)=TPLANPTS(A,B,2)*SIN(DEGREE)+TPLANPTS(A,B,3)*COS(DEGREE)+
            PLANPTS(TRACK(1,1),1,3)*(1-COS(DEGREE))
            -PLANPTS(TRACK(1,1),1,2)*SIN(DEGREE)
6220    NEXT B
6230 NEXT A
```

```
6240 GOTO 3210
6250 '
6260 'Allow another circulation design to be developed
6270 '
6280 ERASE TSPINE,TYARDPTS,TPINPTS,THALLPTS,TPLANPTS
6290 LINE (2,185)-(75,170),0,B:LINE (82,185)-(150,170),0,B:
     LINE(160,185)-(240,170),0,B
6300 LOCATE 23,1:PRINT "DRAW ELEVATIONS?   YES   NO     "
6310 LINE (150,185)-(175,170),1,B:LINE (195,185)-(220,170),1,B
6320 PEN ON
6330 IF PEN(3)=0 THEN GOTO 6320
6340 IF PEN(8)>21 AND PEN(9)<23 THEN GOTO 6690
6350 '
6360 'Allow another circulation design to be developed
6370 '
6380 LINE (2,185)-(75,170),0,B:LINE (82,185)-(150,170),0,B:
     LINE(160,185)-(240,170),0,B
6390 LOCATE 23,1:PRINT "SAVE SITE PLAN?    YES   NO     "
6400 LINE (150,185)-(175,170),1,B:LINE (195,185)-(220,170),1,B
6410 PEN ON
6420 IF PEN(3)=0 THEN GOTO 6410
6430 IF PEN(8)>21 AND PEN(9)<23 THEN JUMP$="SITE":GOTO 6680 ELSE GOTO 6440
6440 LOCATE 23,1:PRINT "ANOTHER SITE?      YES   NO     "
6450 LINE (150,185)-(175,170),1,B:LINE (195,185)-(220,170),1,B
6460 PEN ON
6470 IF PEN(3)=0 THEN GOTO 6460
6480 IF PEN(8)>21 AND PEN(9)<23 THEN GOTO 90
6490 LOCATE 23,1:PRINT "ANOTHER DESIGN?    YES   NO     "
6500 LINE (150,185)-(175,170),1,B:LINE (195,185)-(220,170),1,B
6510 PEN ON
6520 IF PEN(3)=0 THEN GOTO 6510
6530 IF PEN(8)>21 AND PEN(9)>24 THEN 6600
6540 LOCATE 23,1:PRINT "UPDATE DATA?       YES   NO     "
6550 LINE (150,185)-(175,170),1,B:LINE (195,185)-(220,170),1,B
6560 PEN ON
6570 IF PEN(3)=0 THEN GOTO 6560
6580 IF PEN(8)>21 AND PEN(9)>24 THEN GOTO 6670
6590 GOTO 6660
6600 LOCATE 23,1:PRINT "START AGAIN?       YES   NO     "
6610 LINE (150,185)-(175,170),1,B:LINE (195,185)-(220,170),1,B
6620 PEN ON
6630 IF PEN(3)=0 THEN GOTO 6620
6640 IF PEN(8)>21 AND PEN(9)>24 THEN STOP
6650 CHAIN MERGE "CREATE",10,DELETE 10-6640
6660 CHAIN MERGE "SPACE",10,ALL,DELETE 10-6650
6670 CHAIN MERGE "GEOMTRY",10,ALL,DELETE 10-6660
6680 CHAIN MERGE "PLNFILE",10,ALL,DELETE 10-6670
6690 CHAIN MERGE "ELEVTION",10,ALL,DELETE 10-6680
```

```
10 '*************************************************************************
20 'ELEVATION Subroutine
30 '*************************************************************************
40 'This routine allows the user to specify an elevation view and CREATE will
50 'display that view for evaluation and modification.
60 '
70 DIM TSPINE(4,2),TYARDPTS(4,2),TPINPTS(4,2),THALLPTS(12,2),TPLANPTS(20,10,4)
80 ELEVFLAG=1:BASE=50
90 LOCATE 23,1:PRINT SPC(39)
100 LINE (150,185)-(175,170),0,B:LINE (195,185)-(220,170),0,B
110 LOCATE 23,1:INPUT "ENTER ELEVATION  ",ELEV$
120 FLAG=1:COUNT=0
130 WHILE FLAG
140    COUNT=COUNT+1
150    LOCATE 21,1:PRINT "INDICATE ELEVATION ACTIVITIES"
160    FOR B=1 TO 500:NEXT B
170    PEN ON
180    IF PEN(3)=0 THEN GOTO 170
190    X=PEN(4):Y=PEN(5)
200    PSET (X,Y)
210    LOCATE 23,1:PRINT "CORRECT POINT?     YES   NO     "
220    LINE (150,185)-(175,170),1,B:LINE (195,185)-(220,170),1,B
230    PEN ON
240    IF PEN(3)=0 THEN GOTO 230
250    IF PEN(8)>21 AND PEN(9)>24 THEN PSET (X,Y),0:GOTO 160 ELSE GOTO 260
260    PSET (X,Y),0
270    FOR A=1 TO N
280      IF X>PLANPTS(A,1,2) AND X<PLANPTS(A,2,2) AND Y>PLANPTS(A,1,3) AND
          Y<PLANPTS(A,4,3) THEN ACTIVITY=A:GOTO 310 ELSE GOTO 290
290    NEXT A
300    STOP
310    FOR B=1 TO N
320      IF ACTIVITY=TRACK(B,1) THEN GOTO 350
330    NEXT B
340    STOP
350    TEMP(COUNT,1)=ACTIVITY
360    LOCATE 23,1:PRINT "ANOTHER ACTIVITY?  YES   NO     "
370    LINE (150,185)-(175,170),1,B:LINE (195,185)-(220,170),1,B
380    PEN ON
390    IF PEN(3)=0 THEN GOTO 380
400    IF PEN(8)>21 AND PEN(9)>24 THEN FLAG=0
410    LOCATE 23,1:PRINT SPC(39)
420    LINE (150,185)-(175,170),0,B:LINE (195,185)-(220,170),0,B
430 WEND
440 FOR A=1 TO N
450     FOR B=1 TO SPACE(A,11)
460       TPLANPTS(A,B,2)=PLANPTS(A,B,2)
470       TPLANPTS(A,B,3)=PLANPTS(A,B,3)
480     NEXT B
490 NEXT A
500 IF ELEV$="SOUTH" THEN ANGLE=0:GOTO 540 ELSE GOTO 510
510 IF ELEV$="EAST"  THEN ANGLE=.5*3.141593:GOTO 540 ELSE GOTO 520
520 IF ELEV$="NORTH" THEN ANGLE=3.141593:GOTO 540 ELSE GOTO 530
530 IF ELEV$="WEST"  THEN ANGLE=-.5*3.141593:GOTO 540 ELSE STOP
540 IF JUMP=1 THEN GOTO 610
550 IF JUMP=2 THEN GOTO 760
560 IF JUMP=3 THEN GOTO 910
570 IF JUMP=4 THEN GOTO 1140
580 '
590 'Rotate circulation patterns
```

```
600 '
610 FOR A=2 TO 4
620     TSPINE(A,1)=SPINE(A,1)
630     TSPINE(A,2)=SPINE(A,2)
640 NEXT A
650 FOR A=2 TO 4
660     SPINE(A,1)=TSPINE(A,1)*COS(ANGLE)-TSPINE(A,2)*SIN(ANGLE)+
         SPINE(1,1)*(1-COS(ANGLE))+SPINE(1,2)*SIN(ANGLE)
670     SPINE(A,2)=TSPINE(A,1)*SIN(ANGLE)+TSPINE(A,2)*COS(ANGLE)+
         SPINE(1,2)*(1-COS(ANGLE))-SPINE(1,1)*SIN(ANGLE)
680 NEXT A
690 FOR A=1 TO N
700     FOR B=1 TO SPACE(A,11)
710       PLANPTS(A,B,2)=TPLANPTS(A,B,2)*COS(ANGLE)-TPLANPTS(A,B,3)*SIN(ANGLE)+
           SPINE(1,1)*(1-COS(ANGLE))+SPINE(1,2)*SIN(ANGLE)
720       PLANPTS(A,B,3)=TPLANPTS(A,B,2)*SIN(ANGLE)+TPLANPTS(A,B,3)*COS(ANGLE)+
           SPINE(1,2)*(1-COS(ANGLE))-SPINE(1,1)*SIN(ANGLE)
730     NEXT B
740 NEXT A
750 GOTO 1200
760 FOR A=2 TO 4
770     TYARDPTS(A,1)=YARDPTS(A,1)
780     TYARDPTS(A,2)=YARDPTS(A,2)
790 NEXT A
800 FOR A=2 TO 4
810     YARDPTS(A,1)=TYARDPTS(A,1)*COS(ANGLE)-TYARDPTS(A,2)*SIN(ANGLE)+
         YARDPTS(1,1)*(1-COS(ANGLE))+YARDPTS(1,2)*SIN(ANGLE)
820     YARDPTS(A,2)=TYARDPTS(A,1)*SIN(ANGLE)+TYARDPTS(A,2)*COS(ANGLE)+
         YARDPTS(1,2)*(1-COS(ANGLE))-YARDPTS(1,1)*SIN(ANGLE)
830 NEXT A
840 FOR A=1 TO N
850     FOR B=1 TO SPACE(A,11)
860       PLANPTS(A,B,2)=TPLANPTS(A,B,2)*COS(ANGLE)-TPLANPTS(A,B,3)*SIN(ANGLE)+
           YARDPTS(1,1)*(1-COS(ANGLE))+YARDPTS(1,2)*SIN(ANGLE)
870       PLANPTS(A,B,3)=TPLANPTS(A,B,2)*SIN(ANGLE)+TPLANPTS(A,B,3)*COS(ANGLE)+
           YARDPTS(1,2)*(1-COS(ANGLE))-YARDPTS(1,1)*SIN(ANGLE)
880     NEXT B
890 NEXT A
900 GOTO 1200
910 FOR A=2 TO 4
920     TPINPTS(A,1)=PINPTS(A,1)
930     TPINPTS(A,2)=PINPTS(A,2)
940 NEXT A
950 FOR A=2 TO 4
960     PINPTS(A,1)=TPINPTS(A,1)*COS(ANGLE)-TPINPTS(A,2)*SIN(ANGLE)+
         PINPTS(1,1)*(1-COS(ANGLE))+PINPTS(1,2)*SIN(ANGLE)
970     PINPTS(A,2)=TPINPTS(A,1)*SIN(ANGLE)+TPINPTS(A,2)*COS(ANGLE)+
         PINPTS(1,2)*(1-COS(ANGLE))-PINPTS(1,1)*SIN(ANGLE)
980 NEXT A
990 FOR A=1 TO 12
1000     THALLPTS(A,1)=HALLPTS(A,1)
1010     THALLPTS(A,2)=HALLPTS(A,2)
1020 NEXT A
1030 FOR A=1 TO 12
1040     HALLPTS(A,1)=THALLPTS(A,1)*COS(ANGLE)-THALLPTS(A,2)*SIN(ANGLE)+
          HALLPTS(1,1)*(1-COS(ANGLE))+HALLPTS(1,2)*SIN(ANGLE)
1050     HALLPTS(A,2)=THALLPTS(A,1)*SIN(ANGLE)+THALLPTS(A,2)*COS(ANGLE)+
          HALLPTS(1,2)*(1-COS(ANGLE))-HALLPTS(1,1)*SIN(ANGLE)
1060 NEXT A
1070 FOR A=1 TO N
```

```
1080      FOR B=1 TO SPACE(A,11)
1090        PLANPTS(A,B,2)=TPLANPTS(A,B,2)*COS(ANGLE)-TPLANPTS(A,B,3)*SIN(ANGLE)+
             PINPTS(1,1)*(1-COS(ANGLE))+PINPTS(1,2)*SIN(ANGLE)
1100        PLANPTS(A,B,3)=TPLANPTS(A,B,2)*SIN(ANGLE)+TPLANPTS(A,B,3)*COS(ANGLE)+
             PINPTS(1,2)*(1-COS(ANGLE))-PINPTS(1,1)*SIN(ANGLE)
1110      NEXT B
1120    NEXT A
1130    GOTO 1200
1140    FOR A=1 TO N
1150      FOR B=1 TO SPACE(A,11)
1160        PLANPTS(A,B,2)=TPLANPTS(A,B,2)*COS(ANGLE)-TPLANPTS(A,B,3)*SIN(ANGLE)+
               PLANPTS(TRACK(1,1),1,2)*(1-COS(ANGLE))+
               PLANPTS(TRACK(1,1),1,3)*SIN(ANGLE)
1170        PLANPTS(A,B,3)=TPLANPTS(A,B,2)*SIN(ANGLE)+TPLANPTS(A,B,3)*COS(ANGLE)+
               PLANPTS(TRACK(1,1),1,3)*(1-COS(ANGLE))+
               PLANPTS(TRACK(1,1),1,2)*SIN(ANGLE)
1180      NEXT B
1190    NEXT A
1200    '
1210    'Draw elevation view of circulation patterns
1220    '
1230    CLS
1240    IF JUMP=1 THEN GOTO 1280
1250    IF JUMP=2 THEN GOTO 1320
1260    IF JUMP=3 THEN GOTO 1360
1270    IF JUMP=4 THEN GOTO 1420
1280    IF ELEV$="NORTH" OR ELEV$="SOUTH" THEN GOTO 1290 ELSE GOTO 1300
1290    LINE (SPINE(1,1),BASE)-(SPINE(2,1),BASE-10),1,B:GOTO 1420
1300    IF ELEV$="EAST" OR ELEV$="WEST" THEN GOTO 1310 ELSE STOP
1310    LINE (SPINE(1,1),BASE)-(SPINE(4,1),BASE-10),1,B:GOTO 1420
1320    IF ELEV$="NORTH" OR ELEV$="SOUTH" THEN GOTO 1330 ELSE GOTO 1340
1330    LINE (YARDPTS(1,1),BASE)-(YARDPTS(2,1),BASE-10),1,B:GOTO 1420
1340    IF ELEV$="EAST" OR ELEV$="WEST" THEN GOTO 1350 ELSE STOP
1350    LINE (YARDPTS(1,1),BASE)-(YARDPTS(4,1),BASE-10),1,B:GOTO 1420
1360    IF ELEV$="SOUTH" THEN GOTO 1370 ELSE GOTO 1380
1370    LINE (HALLPTS(4,1),BASE)-(PINPTS(3,1),BASE-10),1,B:
        LINE (PINPTS(3,1),BASE)-(HALLPTS(10,1),BASE-10),1,B:
        LINE (HALLPTS(10,1),BASE)-(HALLPTS(2,1),BASE-10),1,B:GOTO 1420
1380    IF ELEV$="NORTH" THEN GOTO 1390 ELSE GOTO 1400
1390    LINE (HALLPTS(1,1),BASE)-(PINPTS(1,1),BASE-10),1,B:
        LINE (PINPTS(1,1),BASE)-(HALLPTS(2,1),BASE-10),1,B:
        LINE (HALLPTS(4,1),BASE)-(HALLPTS(1,1),BASE-10),1,B:GOTO 1420
1400    IF ELEV$="EAST" THEN GOTO 1410 ELSE STOP
1410    LINE (HALLPTS(3,1),BASE)-(PINPTS(3,1),BASE-10),1,B:
        LINE (PINPTS(3,1),BASE)-(HALLPTS(2,1),BASE-10),1,B:
        LINE (HALLPTS(2,1),BASE)-(HALLPTS(1,1),BASE-10),1,B:GOTO 1420
1420    '
1430    'Draw elevations of activities
1440    '
1450    LINE (SITE(1,1),BASE)-(SITE(2,1),BASE),1
1460    FOR A=1 TO COUNT
1470    FLAG=1:COUNTER=0:I=1
1480      WHILE FLAG
1490        I=PLANPTS(TEMP(A,1),I,1)
1500        LINE (PLANPTS(TEMP(A,1),I,2),BASE)-(PLANPTS(TEMP(A,1),PLANPTS
             (TEMP(A,1),I,4),2),BASE),1
1510        LINE (PLANPTS(TEMP(A,1),I,2),BASE-SPACE(TEMP(A,1),4)*SCALE)-
             (PLANPTS(TEMP(A,1),PLANPTS(TEMP(A,1),I,4),2),BASE-
             SPACE(TEMP(A,1),4)*SCALE),1
1520        LINE (PLANPTS(TEMP(A,1),I,2),BASE)-(PLANPTS(TEMP(A,1),I,2),BASE-
```

```
          SPACE(TEMP(A,1),4)*SCALE),1
1530      I=PLANPTS(TEMP(A,1),I,4)
1540      COUNTER=COUNTER+1
1550      IF COUNTER>=SPACE(TEMP(A,1),11) THEN FLAG=0
1560    WEND
1570 NEXT A
1580 LOCATE 9,1:PRINT "ELEVATION VIEW:  ";ELEV$
1590 '
1600 'Draw scale figure
1610 '
1620 LINE (SITE(1,1),BASE)-(SITE(1,1)+1,BASE-2*SCALE),1
1630 LINE (SITE(1,1)+2,BASE)-(SITE(1,1)+1,BASE-2*SCALE),1
1640 CIRCLE (SITE(1,1)+1,BASE-5*SCALE),1
1650 CIRCLE (SITE(1,1)+1,BASE-3*SCALE),1*SCALE
1660 '
1670 'Modify or redraw activities
1680 '
1690 LOCATE 23,1:PRINT "  MODIFY    REDRAW   COMPLETE     "
1700 LINE (2,185)-(75,170),1,B:LINE (82,185)-(150,170),1,B:
     LINE (160,185)-(240,170),1,B
1710 PEN ON
1720 IF PEN(3)=0 THEN GOTO 1710
1730 IF PEN(9)<10 THEN GOTO 1760
1740 IF PEN(9)>=10 AND PEN(9)<22 THEN GOTO 4510
1750 IF PEN(9)>=22 THEN GOTO 3000
1760 LINE (2,185)-(75,170),0,B:LINE (82,185)-(150,170),0,B:
     LINE (160,185)-(240,170),0,B
1770 LOCATE 23,1:PRINT "INDICATE NEW ACTIVITY LOCATION"
1780 FOR A=1 TO 500:NEXT A
1790 PEN ON
1800 IF PEN(3)=0 THEN GOTO 1790
1810 X=PEN(4):Y=PEN(5)
1820 PSET (X,Y)
1830 LOCATE 23,1:PRINT "CORRECT POINT?     YES   NO     "
1840 LINE (150,185)-(175,170),1,B:LINE (195,185)-(220,170),1,B
1850 PEN ON
1860 IF PEN(3)=0 THEN GOTO 1850
1870 IF PEN(8)>21 AND PEN(9)>24 THEN PSET (X,Y),0:GOTO 1780 ELSE GOTO 1880
1880 PSET (X,Y),0
1890 FOR AA=1 TO COUNT
1900 A=TEMP(AA,1)
1910    IF ELEV$="SOUTH" AND X>=PLANPTS(A,1,2) AND X<=PLANPTS(A,2,2)
         THEN ACTIVITY=A:GOTO 1970 ELSE GOTO 1920
1920    IF ELEV$="NORTH" AND X>=PLANPTS(A,3,2) AND X<=PLANPTS(A,4,2)
         THEN ACTIVITY=A:GOTO 1970 ELSE GOTO 1930
1930    IF ELEV$="EAST" AND X>=PLANPTS(A,4,2) AND X<=PLANPTS(A,1,2)
         THEN ACTIVITY=A:GOTO 1970 ELSE GOTO 1940
1940    IF ELEV$="WEST" AND X>=PLANPTS(A,2,2) AND X<=PLANPTS(A,3,2)
         THEN ACTIVITY=A:GOTO 1970 ELSE GOTO 1950
1950 NEXT AA
1960 STOP
1970 FOR B=1 TO 500:NEXT B
1980 PEN ON
1990 IF PEN(3)=0 THEN GOTO 1980
2000 X=PEN(4):Y=PEN(5)
2010 PSET (X,Y)
2020 FOR B=1 TO 500:NEXT B
2030 PEN ON
2040 IF PEN(3)=0 THEN GOTO 2030
2050 IF PEN(8)>21 AND PEN(9)>24 THEN PSET (X,Y),0:GOTO 1970 ELSE GOTO 2060
```

```
2060 PSET (X,Y),0
2070 IF JUMP=1 THEN GOTO 2140
2080 IF JUMP=2 THEN GOTO 2210
2090 IF JUMP=3 THEN GOTO 2560
2100 IF JUMP=4 THEN GOTO 2910
2110 '
2120 'Spine elevations
2130 '
2140 IF ELEV$="NORTH" OR ELEV$="SOUTH" THEN
      XDIST=X-PLANPTS(ACTIVITY,1,2):YDIST=0:GOTO 2160 ELSE GOTO 2150
2150 IF ELEV$="EAST" OR ELEV$="WEST" THEN XDIST=0:YDIST=0:GOTO 2160 ELSE STOP
2160 FOR A=1 TO SPACE(ACTIVITY,11)
2170    PLANPTS(ACTIVITY,A,2)=PLANPTS(ACTIVITY,A,2)+XDIST
2180    PLANPTS(ACTIVITY,A,3)=PLANPTS(ACTIVITY,A,3)+YDIST
2190 NEXT A
2200 GOTO 1200
2210 '
2220 'Courtyard elevations
2230 '
2240 IF ELEV$="SOUTH" THEN GOTO 2280
2250 IF ELEV$="NORTH" THEN GOTO 2350
2260 IF ELEV$="EAST"  THEN GOTO 2420
2270 IF ELEV$="WEST"  THEN GOTO 2490
2280 IF PLANPTS(ACTIVITY,1,3)<YARDPTS(4,2)-1 AND PLANPTS(ACTIVITY,1,3)>
     YARDPTS(4,2)+1 AND PLANPTS(ACTIVITY,2,3)<YARDPTS(4,2)-1 AND PLANPTS
     (ACTIVITY,2,3)>YARDPTS(4,2)+1 THEN XDIST=0:YDIST=0:GOTO 2300 ELSE GOTO 2290

2290 XDIST=X-PLANPTS(ACTIVITY,1,2):YDIST=0
2300 FOR A=1 TO SPACE(ACTIVITY,11)
2310    PLANPTS(ACTIVITY,A,2)=PLANPTS(ACTIVITY,A,2)+XDIST
2320    PLANPTS(ACTIVITY,A,3)=PLANPTS(ACTIVITY,A,3)+YDIST
2330 NEXT A
2340 GOTO 1200
2350 IF PLANPTS(ACTIVITY,4,3)<YARDPTS(1,2)-1 AND PLANPTS(ACTIVITY,4,3)>
     YARDPTS(1,2)+1 AND PLANPTS(ACTIVITY,3,3)<YARDPTS(1,2)-1 AND PLANPTS
     (ACTIVITY,3,3)>PLANPTS(1,2)+1 THEN XDIST=0:YDIST=0:GOTO 2370 ELSE GOTO 2360

2360 XDIST=X-PLANPTS(ACTIVITY,4,2):YDIST=0
2370 FOR A=1 TO SPACE(ACTIVITY,11)
2380    PLANPTS(ACTIVITY,A,2)=PLANPTS(ACTIVITY,A,2)+XDIST
2390    PLANPTS(ACTIVITY,A,3)=PLANPTS(ACTIVITY,A,3)+YDIST
2400 NEXT A
2410 GOTO 1200
2420 IF PLANPTS(ACTIVITY,1,3)<YARDPTS(2,2)-1 AND PLANPTS(ACTIVITY,1,3)
     >YARDPTS(2,2)+1 AND PLANPTS(ACTIVITY,4,3)<YARDPTS(2,2)-1 AND PLANPTS
     (ACTIVITY,4,3)>YARDPTS(2,2)+1 THEN XDIST=0:YDIST=0:GOTO 2440 ELSE GOTO 2430

2430 XDIST=X-PLANPTS(ACTIVITY,4,2):YDIST=0
2440 FOR A=1 TO SPACE(ACTIVITY,11)
2450    PLANPTS(ACTIVITY,A,2)=PLANPTS(ACTIVITY,A,2)+XDIST
2460    PLANPTS(ACTIVITY,A,3)=PLANPTS(ACTIVITY,A,3)+YDIST
2470 NEXT A
2480 GOTO 1200
2490 IF PLANPTS(ACTIVITY,2,3)<YARDPTS(1,2)-1 AND PLANPTS(ACTIVITY,2,3)
     >YARDPTS(1,2)+1 AND PLANPTS(ACTIVITY,3,3)<YARDPTS(1,2)-1 AND PLANPTS
     (ACTIVITY,3,3)>YARDPTS(1,2)+1 THEN XDIST=0:YDIST=0:GOTO 2510 ELSE GOTO 2500

2500 XDIST=X-PLANPTS(ACTIVITY,2,2):YDIST=0
2510 FOR A=1 TO SPACE(ACTIVITY,11)
2520    PLANPTS(ACTIVITY,A,2)=PLANPTS(ACTIVITY,A,2)+XDIST
```

```
2530    PLANPTS(ACTIVITY,A,3)=PLANPTS(ACTIVITY,A,3)+YDIST
2540 NEXT A
2550 GOTO 1200
2560 '
2570 'Pinwheel elevations
2580 '
2590 IF ELEV$="SOUTH" THEN GOTO 2630
2600 IF ELEV$="NORTH" THEN GOTO 2700
2610 IF ELEV$="EAST"  THEN GOTO 2770
2620 IF ELEV$="WEST"  THEN GOTO 2840
2630 IF PLANPTS(ACTIVITY,1,3)<PINPTS(4,2)-1 AND PLANPTS(ACTIVITY,1,3)>
      PINPTS(4,2)+1 AND PLANPTS(ACTIVITY,2,3)<PINPTS(4,2)-1 AND PLANPTS
      (ACTIVITY,2,3)>PINPTS(4,2)+1 THEN XDIST=0:YDIST=0:GOTO 2650 ELSE GOTO 2640

2640 XDIST=X-PLANPTS(ACTIVITY,1,2):YDIST=0
2650 FOR A=1 TO SPACE(ACTIVITY,11)
2660    PLANPTS(ACTIVITY,A,2)=PLANPTS(ACTIVITY,A,2)+XDIST
2670    PLANPTS(ACTIVITY,A,3)=PLANPTS(ACTIVITY,A,3)+YDIST
2680 NEXT A
2690 GOTO 1200
2700 IF PLANPTS(ACTIVITY,4,3)<PINPTS(1,2)-1 AND PLANPTS(ACTIVITY,4,3)>
      PINPTS(1,2)+1 AND PLANPTS(ACTIVITY,3,3)<PINPTS(1,2)-1 AND PLANPTS
      (ACTIVITY,2,3)>PINPTS(1,2)+1 THEN XDIST=0:YDIST=0:GOTO 2390 ELSE GOTO 2710

2710 XDIST=X-PLANPTS(ACTIVITY,4,2):YDIST=0
2720 FOR A=1 TO SPACE(ACTIVITY,11)
2730    PLANPTS(ACTIVITY,A,2)=PLANPTS(ACTIVITY,A,2)+XDIST
2740    PLANPTS(ACTIVITY,A,3)=PLANPTS(ACTIVITY,A,3)+YDIST
2750 NEXT A
2760 GOTO 1200
2770 IF PLANPTS(ACTIVITY,1,3)<PINPTS(2,2)-1 AND PLANPTS(ACTIVITY,1,3)>
      PINPTS(2,2)+1 AND PLANPTS(ACTIVITY,4,3)<PINPTS(2,2)-1 AND PLANPTS
      (ACTIVITY,4,3)>PINPTS(2,2)+1 THEN XDIST=0:YDIST=0:GOTO 2790 ELSE GOTO 2780

2780 XDIST=X-PLANPTS(ACTIVITY,1,2):YDIST=0
2790 FOR A=1 TO SPACE(ACTIVITY,11)
2800    PLANPTS(ACTIVITY,A,2)=PLANPTS(ACTIVITY,A,2)+XDIST
2810    PLANPTS(ACTIVITY,A,3)=PLANPTS(ACTIVITY,A,3)+YDIST
2820 NEXT A
2830 GOTO 1200
2840 IF PLANPTS(ACTIVITY,2,3)<PINPTS(1,2)-1 AND PLANPTS(ACTIVITY,2,3)>
      PINPTS(1,2)+1 AND PLANPTS(ACTIVITY,3,3)<PINPTS(1,2)-1 AND PLANPTS
      (ACTIVITY,3,3)>PINPTS(1,2)+1 THEN XDIST=0:YDIST=0:GOTO 2860 ELSE GOTO 2850

2850 XDIST=X-PLANPTS(ACTIVITY,2,2):YDIST=0
2860 FOR A=1 TO SPACE(ACTIVITY,11)
2870    PLANPTS(ACTIVITY,A,2)=PLANPTS(ACTIVITY,A,2)+XDIST
2880    PLANPTS(ACTIVITY,A,3)=PLANPTS(ACTIVITY,A,3)+YDIST
2890 NEXT A
2900 GOTO 1200
2910 '
2920 'Cluster elevations
2930 '
2940 XDIST=X-PLANPTS(ACTIVITY,1,2):YDIST=Y-PLANPTS(ACTIVITY,1,3)
2950 FOR A=1 TO SPACE(ACTIVITY,11)
2960    PLANPTS(ACTIVITY,A,2)=PLANPTS(ACTIVITY,A,2)+XDIST
2970    PLANPTS(ACTIVITY,A,3)=PLANPTS(ACTIVITY,A,3)+YDIST
2980 NEXT A
2990 GOTO 1200
3000 'LOCATE 23,1:PRINT "ANOTHER ELEVATION? YES   NO      "
```

```
3010 'LINE (150,185)-(175,170),1,B:LINE (195,185)-(220,170),1,B
3020 'PEN ON
3030 'IF PEN(3)=0 THEN GOTO 1670
3040 'IF PEN(8)>21 AND PEN(9)<23 THEN GOTO 10
3050 '
3060 'Store plan and site in temporary files
3070 '
3080 FOR A=1 TO N
3090    FOR B=1 TO SPACE(A,11)
3100      TPLANPTS(A,B,2)=PLANPTS(A,B,2)
3110      TPLANPTS(A,B,3)=PLANPTS(A,B,3)
3120    NEXT B
3130 NEXT A
3140 IF ELEV$="SOUTH" THEN ANGLE=0:GOTO 3180 ELSE GOTO 3150
3150 IF ELEV$="EAST"  THEN ANGLE=-.5*3.141593:GOTO 3180 ELSE GOTO 3160
3160 IF ELEV$="NORTH" THEN ANGLE=-3.141593:GOTO 3180 ELSE GOTO 3170
3170 IF ELEV$="WEST"  THEN ANGLE=.5*3.141593:GOTO 3180 ELSE STOP
3180 IF JUMP=1 THEN GOTO 3220
3190 IF JUMP=2 THEN GOTO 3370
3200 IF JUMP=3 THEN GOTO 3520
3210 IF JUMP=4 THEN GOTO 3750
3220 FOR A=2 TO 4
3230    TSPINE(A,1)=SPINE(A,1)
3240    TSPINE(A,2)=SPINE(A,2)
3250 NEXT A
3260 FOR A=2 TO 4
3270    SPINE(A,1)=TSPINE(A,1)*COS(ANGLE)-TSPINE(A,2)*SIN(ANGLE)+
         SPINE(1,1)*(1-COS(ANGLE))+SPINE(1,2)*SIN(ANGLE)
3280    SPINE(A,2)=TSPINE(A,1)*SIN(ANGLE)+TSPINE(A,2)*COS(ANGLE)+
         SPINE(1,2)*(1-COS(ANGLE))-SPINE(1,1)*SIN(ANGLE)
3290 NEXT A
3300 FOR A=1 TO N
3310    FOR B=1 TO SPACE(A,11)
3320      PLANPTS(A,B,2)=TPLANPTS(A,B,2)*COS(ANGLE)-TPLANPTS(A,B,3)*SIN(ANGLE)+
           SPINE(1,1)*(1-COS(ANGLE))+SPINE(1,2)*SIN(ANGLE)
3330      PLANPTS(A,B,3)=TPLANPTS(A,B,2)*SIN(ANGLE)+TPLANPTS(A,B,3)*COS(ANGLE)+
           SPINE(1,2)*(1-COS(ANGLE))-SPINE(1,1)*SIN(ANGLE)
3340    NEXT B
3350 NEXT A
3360 GOTO 3810
3370 FOR A=2 TO 4
3380    TYARDPTS(A,1)=YARDPTS(A,1)
3390    TYARDPTS(A,2)=YARDPTS(A,2)
3400 NEXT A
3410 FOR A=2 TO 4
3420    YARDPTS(A,1)=TYARDPTS(A,1)*COS(ANGLE)-TYARDPTS(A,2)*SIN(ANGLE)+
         YARDPTS(1,1)*(1-COS(ANGLE))+YARDPTS(1,2)*SIN(ANGLE)
3430    YARDPTS(A,2)=TYARDPTS(A,1)*SIN(ANGLE)+TYARDPTS(A,2)*COS(ANGLE)+
         YARDPTS(1,2)*(1-COS(ANGLE))-YARDPTS(1,1)*SIN(ANGLE)
3440 NEXT A
3450 FOR A=1 TO N
3460    FOR B=1 TO SPACE(A,11)
3470      PLANPTS(A,B,2)=TPLANPTS(A,B,2)*COS(ANGLE)-TPLANPTS(A,B,3)*SIN(ANGLE)+
           YARDPTS(1,1)*(1-COS(ANGLE))+YARDPTS(1,2)*SIN(ANGLE)
3480      PLANPTS(A,B,3)=TPLANPTS(A,B,2)*SIN(ANGLE)+TPLANPTS(A,B,3)*COS(ANGLE)+
           YARDPTS(1,2)*(1-COS(ANGLE))-YARDPTS(1,1)*SIN(ANGLE)
3490    NEXT B
3500 NEXT A
3510 GOTO 3810
3520 FOR A=2 TO 4
```

```
3530     TPINPTS(A,1)=PINPTS(A,1)
3540     TPINPTS(A,2)=PINPTS(A,2)
3550 NEXT A
3560 FOR A=2 TO 4
3570     PINPTS(A,1)=TPINPTS(A,1)*COS(ANGLE)-TPINPTS(A,2)*SIN(ANGLE)+
          PINPTS(1,1)*(1-COS(ANGLE))+PINPTS(1,2)*SIN(ANGLE)
3580     PINPTS(A,2)=TPINPTS(A,1)*SIN(ANGLE)+TPINPTS(A,2)*COS(ANGLE)+
          PINPTS(1,2)*(1-COS(ANGLE))-PINPTS(1,1)*SIN(ANGLE)
3590 NEXT A
3600 FOR A=1 TO 12
3610     THALLPTS(A,1)=HALLPTS(A,1)
3620     THALLPTS(A,2)=HALLPTS(A,2)
3630 NEXT A
3640 FOR A=1 TO 12
3650     HALLPTS(A,1)=THALLPTS(A,1)*COS(ANGLE)-THALLPTS(A,2)*SIN(ANGLE)+
          HALLPTS(1,1)*(1-COS(ANGLE))+HALLPTS(1,2)*SIN(ANGLE)
3660     HALLPTS(A,2)=THALLPTS(A,1)*SIN(ANGLE)+THALLPTS(A,2)*COS(ANGLE)+
          HALLPTS(1,2)*(1-COS(ANGLE))-HALLPTS(1,1)*SIN(ANGLE)
3670 NEXT A
3680 FOR A=1 TO N
3690     FOR B=1 TO SPACE(A,11)
3700       PLANPTS(A,B,2)=TPLANPTS(A,B,2)*COS(ANGLE)-TPLANPTS(A,B,3)*SIN(ANGLE)+
            PINPTS(1,1)*(1-COS(ANGLE))+PINPTS(1,2)*SIN(ANGLE)
3710       PLANPTS(A,B,3)=TPLANPTS(A,B,2)*SIN(ANGLE)+TPLANPTS(A,B,3)*COS(ANGLE)+
            PINPTS(1,2)*(1-COS(ANGLE))-PINPTS(1,1)*SIN(ANGLE)
3720     NEXT B
3730 NEXT A
3740 GOTO 3810
3750 FOR A=1 TO N
3760     FOR B=1 TO SPACE(A,11)
3770       PLANPTS(A,B,2)=TPLANPTS(A,B,2)*COS(ANGLE)-TPLANPTS(A,B,3)*SIN(ANGLE)+
              PLANPTS(TRACK(1,1),1,2)*(1-COS(ANGLE))+
              PLANPTS(TRACK(1,1),1,3)*SIN(ANGLE)
3780       PLANPTS(A,B,3)=TPLANPTS(A,B,2)*SIN(ANGLE)+TPLANPTS(A,B,3)*COS(ANGLE)+
              PLANPTS(TRACK(1,1),1,3)*(1-COS(ANGLE))+
              PLANPTS(TRACK(1,1),1,2)*SIN(ANGLE)
3790     NEXT B
3800 NEXT A
3810 '
3820 LOCATE 23,1:PRINT SPC(39)
3830 LINE (2,185)-(75,170),0,B:LINE (82,185)-(150,170),0,B:
      LINE (160,185)-(240,170),0,B
3840 LOCATE 23,1:PRINT "PLAN ENTERED INTO B:PLAN"
3850 FOR A=1 TO 1000:NEXT A
3860 LOCATE 23,1:PRINT "SITE ENTERED INTO B:SITE"
3870 FILENM1$="B:PLAN":FILENM2$="B:SITE"
3880 OPEN FILENM1$ FOR OUTPUT AS #1
3890 PRINT#1,JUMP
3900 PRINT#1,N
3910 IF JUMP=1 THEN GOTO 3950
3920 IF JUMP=2 THEN GOTO 4000
3930 IF JUMP=3 THEN GOTO 4050
3940 IF JUMP=4 THEN GOTO 4130
3950 FOR A=1 TO 4
3960     PRINT#1,SPINE(A,1)
3970     PRINT#1,SPINE(A,2)
3980 NEXT A
3990 GOTO 4130
4000 FOR A=1 TO 4
4010     PRINT#1,YARDPTS(A,1)
```

```
4020     PRINT#1,YARDPTS(A,2)
4030 NEXT A
4040 GOTO 4130
4050 FOR A=1 TO 4
4060     PRINT#1,PINPTS(A,1)
4070     PRINT#1,PINPTS(A,2)
4080 NEXT A
4090 FOR A=1 TO 12
4100     PRINT#1,HALLPTS(A,1)
4110     PRINT#1,HALLPTS(A,2)
4120 NEXT A
4130 FOR A=1 TO N
4140     PRINT#1,SPACE(A,4)
4150     PRINT#1,SPACE(A,11)
4160 NEXT A
4170 FOR A=1 TO N
4180     PRINT#1,TRACK(A,1)
4190 NEXT A
4200 FOR A=1 TO N
4210     FOR B=1 TO SPACE(A,11)
4220       PRINT#1,PLANPTS(A,B,1)
4230       PRINT#1,PLANPTS(A,B,2)
4240       PRINT#1,PLANPTS(A,B,3)
4250       PRINT#1,PLANPTS(A,B,4)
4260     NEXT B
4270 NEXT A
4280 FOR A=1 TO N
4290     PRINT#1,TITLE$(A,A)
4300 NEXT A
4310 CLOSE#1
4320 OPEN FILENM2$ FOR OUTPUT AS #2
4330 PRINT#2,NUMSITEPOINTS
4340 PRINT#2,NUMEASEPOINTS
4350 PRINT#2,NUMTREES
4360 FOR A=1 TO NUMSITEPOINTS
4370     PRINT#2,SITE(A,1)
4380     PRINT#2,SITE(A,2)
4390 NEXT A
4400 FOR A=1 TO NUMEASEPOINTS
4410     PRINT#2,EASE(A,1)
4420     PRINT#2,EASE(A,2)
4430 NEXT A
4440 FOR A=1 TO NUMTREES
4450     PRINT#2,TREE(A,1)
4460     PRINT#2,TREE(A,2)
4470     PRINT#2,TREE(A,3)
4480 NEXT A
4490 CLOSE #2
4500 GOTO 4830
4510 '
4520 'Redraw activities
4530 '
4540 LOCATE 23,1:PRINT SPC(39)
4550 LINE (2,185)-(75,170),0,B:LINE (82,185)-(150,170),0,B:
      LINE (160,185)-(240,170),0,B
4560 IF ELEV$<>"SOUTH" THEN GOTO 4700
4570 FOR A=1 TO COUNT
4580 FLAG=1:COUNTER=0:I=1
4590     WHILE FLAG
4600       I=PLANPTS(TEMP(A,1),I,1)
```

```
4610       LINE (PLANPTS(TEMP(A,1),I,2),BASE)-(PLANPTS(TEMP(A,1),PLANPTS(TEMP(A,1
           ),I,4),2),BASE),1
4620       LINE (PLANPTS(TEMP(A,1),I,2),BASE-SPACE(TEMP(A,1),4)*SCALE)-(PLANPTS(
           TEMP(A,1),PLANPTS(TEMP(A,1),I,4),2),BASE-SPACE(TEMP(A,1),4)*SCALE),1
4630       LINE (PLANPTS(TEMP(A,1),I,2),BASE)-(PLANPTS(TEMP(A,1),I,2),BASE-
           SPACE(TEMP(A,1),4)*SCALE),1
4640       I=PLANPTS(TEMP(A,1),I,4)
4650       COUNTER=COUNTER+1
4660       IF COUNTER>=SPACE(TEMP(A,1),11) THEN FLAG=0
4670     WEND
4680 NEXT A
4690 GOTO 1620
4700 FOR A=1 TO COUNT
4710 FLAG=1:COUNTER=0:I=1
4720     WHILE FLAG
4730       I=PLANPTS(TEMP(A,1),I,1)
4740       LINE (TEMPTS(TEMP(A,1),I,2),BASE)-(TEMPTS(TEMP(A,1),TEMPTS(TEMP(A,1),I
           ,4),2),BASE),1
4750       LINE (TEMPTS(TEMP(A,1),I,2),BASE-SPACE(TEMP(A,1),4)*SCALE)-(TEMPTS(
           TEMP(A,1),TEMPTS(TEMP(A,1),I,4),2),BASE-SPACE(TEMP(A,1),4)*SCALE),1
4760       LINE (TEMPTS(TEMP(A,1),I,2),BASE)-(TEMPTS(TEMP(A,1),I,2),BASE-
           SPACE(TEMP(A,1),4)*SCALE),1
4770       I=TEMPTS(TEMP(A,1),I,4)
4780       COUNTER=COUNTER+1
4790       IF COUNTER>=SPACE(TEMP(A,1),11) THEN FLAG=0
4800     WEND
4810 NEXT A
4820 GOTO 1620
4830 ERASE TSPINE,TYARDPTS,TPINPTS,THALLPTS,TPLANPTS
4840 CHAIN MERGE "PICTURE",10,ALL,DELETE 10-4830
```

```
10 '*****************************************************************************
20 'PICTURE Subroutine
30 '*****************************************************************************
31 'This routine allows the user to modify site plans and building plans.
32 'The modified plans may then be placed back on a site plan or the user
33 'may develop elevation views to analyze the building form.
34 '
40 DIM TSPINE(4,2),TYARDPTS(4,2),TPINPTS(4,2),THALLPTS(12,2),TPLANPTS(20,10,4)
50 CLS
60 LOCATE 1,12:PRINT "PICTURE Subroutine"
70 LOCATE 4,1
80 IF ELEVFLAG=1 THEN DATANM$="B:PLAN":GOTO 120 ELSE GOTO 90
90 LOCATE 3,1:PRINT "PLANS ON DISK"
100 FILES "B:"
110 LOCATE 23,1:INPUT "ENTER FILE NAME FOR PLAN: ",DATANM$
120 CLS
130 OPEN DATANM$ FOR INPUT AS #1
140 INPUT#1,JUMP
150 INPUT#1,N
160 IF JUMP=1 THEN GOTO 200
170 IF JUMP=2 THEN GOTO 250
180 IF JUMP=3 THEN GOTO 300
190 IF JUMP=4 THEN GOTO 380
200 FOR A=1 TO 4
210     INPUT#1,SPINE(A,1)
220     INPUT#1,SPINE(A,2)
230 NEXT A
240 GOTO 380
250 FOR A=1 TO 4
260     INPUT#1,YARDPTS(A,1)
270     INPUT#1,YARDPTS(A,2)
280 NEXT A
290 GOTO 380
300 FOR A=1 TO 4
310    INPUT#1,PINPTS(A,1)
320    INPUT#1,PINPTS(A,2)
330 NEXT A
340 FOR A=1 TO 12
350    INPUT#1,HALLPTS(A,1)
360    INPUT#1,HALLPTS(A,2)
370 NEXT A
380 FOR A=1 TO N
390    INPUT#1,SPACE(A,4)
400    INPUT#1,SPACE(A,11)
410 NEXT A
420 FOR A=1 TO N
430    INPUT#1,TRACK(A,1)
440 NEXT A
450 FOR A=1 TO N
460    FOR B=1 TO SPACE(A,11)
470      INPUT#1,PLANPTS(A,B,1)
480      INPUT#1,PLANPTS(A,B,2)
490      INPUT#1,PLANPTS(A,B,3)
500      INPUT#1,PLANPTS(A,B,4)
510    NEXT B
520 NEXT A
530 FOR A=1 TO N
540     INPUT#1,TITLE$(A,A)
550 NEXT A
560 CLOSE#1
```

```
570 IF ELEVFLAG=1 THEN ELEVFLAG=0:DATANM1$="B:SITE":GOTO 590 ELSE GOTO 580
580 LOCATE 23,1:INPUT "ENTER FILE NAME FOR SITE: ",DATANM1$
590 OPEN DATANM1$ FOR INPUT AS #2
600 INPUT#2,NUMSITEPOINTS
610 INPUT#2,NUMEASEPOINTS
620 INPUT#2,NUMTREES
630 FOR A=1 TO NUMSITEPOINTS
640     INPUT#2,SITE(A,1)
650     INPUT#2,SITE(A,2)
660 NEXT A
670 FOR A=1 TO NUMEASEPOINTS
680     INPUT#2,EASE(A,1)
690     INPUT#2,EASE(A,2)
700 NEXT A
710 FOR A=1 TO NUMTREES
720     INPUT#2,TREE(A,1)
730     INPUT#2,TREE(A,2)
740     INPUT#2,TREE(A,3)
750 NEXT A
760 CLOSE#2
770 FOR A=1 TO NUMSITEPOINTS-1
780     LINE (SITE(A,1),SITE(A,2))-(SITE(A+1,1),SITE(A+1,2)),1
790 NEXT A
800 LINE (SITE(A,1),SITE(A,2))-(SITE(1,1),SITE(1,2)),1
810 IF NUMEASEPOINTS=0 THEN GOTO 880
820 FOR A=1 TO NUMEASEPOINTS-1
830     LINE (EASE(A,1),EASE(A,2))-(EASE(A+1,1),EASE(A+1,2)),1
840 NEXT A
850 LINE (EASE (A,1),EASE(A,2))-(EASE(1,1),EASE(1,2)),1
860 PAINT (1,1),3,1
870 IF NUMTREES=0 THEN GOTO 920
880 FOR A=1 TO NUMTREES
890     CIRCLE(TREE(A,1),TREE(A,2)),TREE(A,3),1
900     PAINT (TREE(A,1),TREE(A,2)),3,1
910 NEXT A
920 LOCATE 22,37:PRINT "N"
930 LINE (285,160)-(300,160),1:LINE (300,160)-(292,152),1:
    LINE (292,152)-(285,160),1
940 PAINT (292,155),3,1
950 '
960 'Draw circulation pattern and activities
970 '
980 IF JUMP=1 THEN GOTO 1020
990 IF JUMP=2 THEN GOTO 1070
1000 IF JUMP=3 THEN GOTO 1120
1010 IF JUMP=4 THEN GOTO 1210
1020 '
1030 'Draw spine and activities
1040 '
1050 LINE (SPINE(1,1),SPINE(1,2))-(SPINE(3,1),SPINE(3,2)),1,B
1060 GOTO 1240
1070 '
1080 'Draw courtyard and activities
1090 '
1100 LINE (YARDPTS(1,1),YARDPTS(1,2))-(YARDPTS(3,1),YARDPTS(3,2)),1,B
1110 GOTO 1240
1120 '
1130 'Draw pinwheel and activities
1140 '
1150 LINE (PINPTS(1,1),PINPTS(1,2))-(PINPTS(3,1),PINPTS(3,2)),1,B
```

```
1160 LINE (HALLPTS(1,1),HALLPTS(1,2))-(PINPTS(4,1),PINPTS(4,2)),1,B
1170 LINE (HALLPTS(2,1),HALLPTS(2,2))-(PINPTS(1,1),PINPTS(1,2)),1,B
1180 LINE (HALLPTS(3,1),HALLPTS(3,2))-(PINPTS(2,1),PINPTS(2,2)),1,B
1190 LINE (HALLPTS(4,1),HALLPTS(4,2))-(PINPTS(3,1),PINPTS(3,2)),1,B
1200 GOTO 1240
1210 '
1220 'Draw activities
1230 '
1240 FOR A=1 TO N
1250 FLAG=1:I=1:COUNT=0
1260 WHILE FLAG
1270    I=PLANPTS(TRACK(A,1),I,1)
1280    LINE (PLANPTS(TRACK(A,1),PLANPTS(TRACK(A,1),I,1),2),PLANPTS(TRACK(A,1),
         PLANPTS(TRACK(A,1),I,1),3))-(PLANPTS(TRACK(A,1),PLANPTS(TRACK(A,1),I,4
         ),2),PLANPTS(TRACK(A,1),PLANPTS(TRACK(A,1),I,4),3)),1
1290    I=PLANPTS(TRACK(A,1),I,4)
1300    COUNT=COUNT+1
1310    IF COUNT>=SPACE(TRACK(A,1),11) THEN FLAG=0
1320 WEND
1330 NEXT A
1340 FOR A=1 TO N
1350 XPLACE=INT((PLANPTS(TRACK(A,1),1,2)+(PLANPTS(TRACK(A,1),2,2)-
      PLANPTS(TRACK(A,1),1,2))/2)/8)
1360 YPLACE=INT((PLANPTS(TRACK(A,1),1,3)+(PLANPTS(TRACK(A,1),4,3)-
      PLANPTS(TRACK(A,1),1,3))/2)/8)
1370    LOCATE YPLACE,XPLACE
1380    IF A<10 THEN PRINT USING "#";TRACK(A,1)
1390    IF A>=10 THEN PRINT USING "##";TRACK(A,1)
1400 IF N<=6 THEN GOTO 1430
1410 IF N<=12 THEN GOTO 1460
1420 IF N<=20 THEN GOTO 1490
1430 LOCATE 13+A,1:PRINT USING "#";TRACK(A,1)
1440 LOCATE 13+A,3:PRINT USING "\         \";TITLE$(TRACK(A,1),TRACK(A,1))
1450 GOTO 1510
1460 LOCATE 7+A,14:PRINT USING "#";TRACK(A,1)
1470 LOCATE 7+A,17:PRINT USING "\         \";TITLE$(TRACK(A,1),TRACK(A,1))
1480 GOTO 1510
1490 LOCATE A,28:PRINT USING "#";TRACK(A,1)
1500 LOCATE A,31:PRINT USING "\         \";TITLE$(TRACK(A,1),TRACK(A,1))
1510 NEXT A
1520 '
1530 'Modify site or plan drawings
1540 '
1550 LOCATE 20,1:PRINT " RELOCATE   ROTATE                  "
1560 LINE (2,165)-(75,150),1,B:LINE (82,165)-(150,150),1,B:
1570 LOCATE 23,1:PRINT "  MODIFY    REDRAW   COMPLETE      "
1580 LINE (2,185)-(75,170),1,B:LINE (82,185)-(150,170),1,B:
      LINE(160,185)-(240,170),1,B
1590 PEN ON
1600 IF PEN(3)=0 THEN GOTO 1590
1610 IF PEN(8)>19 AND PEN(8)<22 AND PEN(9)<12 THEN GOTO 1660
1620 IF PEN(8)>19 AND PEN(8)<22 AND PEN(9)>=12 THEN GOTO 2730
1630 IF PEN(8)>=22 AND PEN(9)<12 THEN GOTO 4510
1640 IF PEN(8)>=22 AND PEN(9)>=12 AND PEN(9)<20 THEN GOTO 5540
1650 IF PEN(8)>=22 AND PEN(9)>=20 THEN GOTO 5850 ELSE STOP
1660 '
1670 'Translation of building
1680 '
1690 LINE (2,185)-(75,170),0,B:LINE (82,185)-(150,170),0,B:
      LINE(160,185)-(240,170),0,B
```

```
1700 LOCATE 23,1:PRINT SPC(39)
1710 LOCATE 23,1:PRINT "INDICATE NEW LOCATION            "
1720 FOR A=1 TO 500:NEXT A
1730 PEN ON
1740 IF PEN(3)=0 THEN GOTO 1730
1750 X=PEN(4):Y=PEN(5)
1760 PSET (X,Y)
1770 LOCATE 23,1:PRINT "CORRECT POINT?     YES   NO     "
1780 LINE (150,185)-(175,170),1,B:LINE (195,185)-(220,170),1,B
1790 PEN ON
1800 IF PEN(3)=0 THEN GOTO 1790
1810 IF PEN(8)>21 AND PEN(9)>24 THEN PSET (X,Y),0:GOTO 1720 ELSE GOTO 1820
1820 PSET (X,Y),0
1830 LINE (150,185)-(175,170),0,B:LINE (195,185)-(220,170),0,B
1840 LOCATE 23,1:PRINT SPC(39)
1850 LINE (150,185)-(175,170),0,B:LINE (195,185)-(220,170),0,B
1860 IF JUMP=1 THEN GOTO 1900
1870 IF JUMP=2 THEN GOTO 2040
1880 IF JUMP=3 THEN GOTO 2180
1890 IF JUMP=4 THEN GOTO 2520 ELSE STOP
1900 XDIST=X-SPINE(1,1):YDIST=Y-SPINE(1,2)
1910 LINE (SPINE(1,1),SPINE(1,2))-(SPINE(2,1),SPINE(2,2)),0
1920 LINE (SPINE(2,1),SPINE(2,2))-(SPINE(3,1),SPINE(3,2)),0
1930 LINE (SPINE(3,1),SPINE(3,2))-(SPINE(4,1),SPINE(4,2)),0
1940 LINE (SPINE(4,1),SPINE(4,2))-(SPINE(1,1),SPINE(1,2)),0
1950 FOR A=1 TO 4
1960    SPINE(A,1)=SPINE(A,1)+XDIST
1970    SPINE(A,2)=SPINE(A,2)+YDIST
1980 NEXT A
1990 LINE (SPINE(1,1),SPINE(1,2))-(SPINE(2,1),SPINE(2,2)),1
2000 LINE (SPINE(2,1),SPINE(2,2))-(SPINE(3,1),SPINE(3,2)),1
2010 LINE (SPINE(3,1),SPINE(3,2))-(SPINE(4,1),SPINE(4,2)),1
2020 LINE (SPINE(4,1),SPINE(4,2))-(SPINE(1,1),SPINE(1,2)),1
2030 GOTO 2530
2040 XDIST=X-YARDPTS(1,1):YDIST=Y-YARDPTS(1,2)
2050 LINE (YARDPTS(1,1),YARDPTS(1,2))-(YARDPTS(2,1),YARDPTS(2,2)),0
2060 LINE (YARDPTS(2,1),YARDPTS(2,2))-(YARDPTS(3,1),YARDPTS(3,2)),0
2070 LINE (YARDPTS(3,1),YARDPTS(3,2))-(YARDPTS(4,1),YARDPTS(4,2)),0
2080 LINE (YARDPTS(4,1),YARDPTS(4,2))-(YARDPTS(1,1),YARDPTS(1,2)),0
2090 FOR A=1 TO 4
2100    YARDPTS(A,1)=YARDPTS(A,1)+XDIST
2110    YARDPTS(A,2)=YARDPTS(A,2)+YDIST
2120 NEXT A
2130 LINE (YARDPTS(1,1),YARDPTS(1,2))-(YARDPTS(2,1),YARDPTS(2,2)),1
2140 LINE (YARDPTS(2,1),YARDPTS(2,2))-(YARDPTS(3,1),YARDPTS(3,2)),1
2150 LINE (YARDPTS(3,1),YARDPTS(3,2))-(YARDPTS(4,1),YARDPTS(4,2)),1
2160 LINE (YARDPTS(4,1),YARDPTS(4,2))-(YARDPTS(1,1),YARDPTS(1,2)),1
2170 GOTO 2530
2180 XDIST=X-PINPTS(1,1):YDIST=Y-PINPTS(1,2)
2190 LINE (HALLPTS(12,1),HALLPTS(12,2))-(HALLPTS(1,1),HALLPTS(1,2)),0
2200 LINE (HALLPTS(1,1),HALLPTS(1,2))-(HALLPTS(5,1),HALLPTS(5,2)),0
2210 LINE (HALLPTS(5,1),HALLPTS(5,2))-(PINPTS(4,1),PINPTS(4,2)),0
2220 LINE (HALLPTS(9,1),HALLPTS(9,2))-(HALLPTS(2,1),HALLPTS(2,2)),0
2230 LINE (HALLPTS(2,1),HALLPTS(2,2))-(HALLPTS(6,1),HALLPTS(6,2)),0
2240 LINE (HALLPTS(6,1),HALLPTS(6,2))-(PINPTS(1,1),PINPTS(1,2)),0
2250 LINE (PINPTS(2,1),PINPTS(2,2))-(HALLPTS(7,1),HALLPTS(7,2)),0
2260 LINE (HALLPTS(7,1),HALLPTS(7,2))-(HALLPTS(3,1),HALLPTS(3,2)),0
2270 LINE (HALLPTS(3,1),HALLPTS(3,2))-(HALLPTS(10,1),HALLPTS(10,2)),0
2280 LINE (PINPTS(3,1),PINPTS(3,2))-(HALLPTS(8,1),HALLPTS(8,2)),0
2290 LINE (HALLPTS(8,1),HALLPTS(8,2))-(HALLPTS(4,1),HALLPTS(4,2)),0
```

```
2300 LINE (HALLPTS(4,1),HALLPTS(4,2))-(HALLPTS(11,1),HALLPTS(11,2)),0
2310 FOR A=1 TO 4
2320    PINPTS(A,1)=PINPTS(A,1)+XDIST
2330    PINPTS(A,2)=PINPTS(A,2)+YDIST
2340 NEXT A
2350 FOR A=1 TO 12
2360    HALLPTS(A,1)=HALLPTS(A,1)+XDIST
2370    HALLPTS(A,2)=HALLPTS(A,2)+YDIST
2380 NEXT A
2390 LINE (HALLPTS(12,1),HALLPTS(12,2))-(HALLPTS(1,1),HALLPTS(1,2)),1
2400 LINE (HALLPTS(1,1),HALLPTS(1,2))-(HALLPTS(5,1),HALLPTS(5,2)),1
2410 LINE (HALLPTS(5,1),HALLPTS(5,2))-(PINPTS(4,1),PINPTS(4,2)),1
2420 LINE (HALLPTS(9,1),HALLPTS(9,2))-(HALLPTS(2,1),HALLPTS(2,2)),1
2430 LINE (HALLPTS(2,1),HALLPTS(2,2))-(HALLPTS(6,1),HALLPTS(6,2)),1
2440 LINE (HALLPTS(6,1),HALLPTS(6,2))-(PINPTS(1,1),PINPTS(1,2)),1
2450 LINE (PINPTS(2,1),PINPTS(2,2))-(HALLPTS(7,1),HALLPTS(7,2)),1
2460 LINE (HALLPTS(7,1),HALLPTS(7,2))-(HALLPTS(3,1),HALLPTS(3,2)),1
2470 LINE (HALLPTS(3,1),HALLPTS(3,2))-(HALLPTS(10,1),HALLPTS(10,2)),1
2480 LINE (PINPTS(3,1),PINPTS(3,2))-(HALLPTS(8,1),HALLPTS(8,2)),1
2490 LINE (HALLPTS(8,1),HALLPTS(8,2))-(HALLPTS(4,1),HALLPTS(4,2)),1
2500 LINE (HALLPTS(4,1),HALLPTS(4,2))-(HALLPTS(11,1),HALLPTS(11,2)),1
2510 GOTO 2530
2520 XDIST=X-PLANPTS(1,1,2):YDIST=Y-PLANPTS(1,1,3)
2530 '
2540 'Translate to new coordinates
2550 '
2551 FOR A=1 TO N
2552    XPLACE=INT((PLANPTS(TRACK(A,1),1,2)+(PLANPTS(TRACK(A,1),2,2)-
        PLANPTS(TRACK(A,1),1,2))/2)/8)
2553    YPLACE=INT((PLANPTS(TRACK(A,1),1,3)+(PLANPTS(TRACK(A,1),4,3)-
        PLANPTS(TRACK(A,1),1,3))/2)/8)
2554    LOCATE YPLACE,XPLACE
2555    IF A<10 THEN PRINT " "
2556    IF A>=10 THEN PRINT "  "
2557 NEXT A
2560 FOR A=1 TO N
2570 FLAG=1:I=1:COUNT=0
2580 WHILE FLAG
2590    I=PLANPTS(TRACK(A,1),I,1)
2600    LINE (PLANPTS(TRACK(A,1),PLANPTS(TRACK(A,1),I,1),2),PLANPTS(TRACK(A,1),
        PLANPTS(TRACK(A,1),I,1),3))-(PLANPTS(TRACK(A,1),PLANPTS(TRACK(A,1),I,4)
        ,2),PLANPTS(TRACK(A,1),PLANPTS(TRACK(A,1),I,4).3)),0
2610    I=PLANPTS(TRACK(A,1),I,4)
2620    COUNT=COUNT+1
2630    IF COUNT>=SPACE(TRACK(A,1),11) THEN FLAG=0
2640 WEND
2650 NEXT A
2660 FOR A=1 TO N
2670    FOR B=1 TO SPACE(A,11)
2680      PLANPTS(A,B,2)=PLANPTS(A,B,2)+XDIST
2690      PLANPTS(A,B,3)=PLANPTS(A,B,3)+YDIST
2700    NEXT B
2710 NEXT A
2720 GOTO 1210
2730 '
2740 'Rotation of building
2750 '
2760 FOR A=1 TO N
2770    FOR B=1 TO SPACE(A,11)
2780      TPLANPTS(A,B,2)=PLANPTS(A,B,2)
```

```
2790       TPLANPTS(A,B,3)=PLANPTS(A,B,3)
2800     NEXT B
2810 NEXT A
2820 LINE (2,185)-(75,170),0,B:LINE (82,185)-(150,170),0,B:
      LINE(160,185)-(240,170),0,B
2830 LOCATE 23,1:PRINT SPC(39)
2840 LOCATE 23,1:INPUT "ENTER DEGREES ROTATION  ",DEGREE
2850 DEGREE=DEGREE*(-3.141593/180)
2860 IF JUMP=1 THEN GOTO 2900
2870 IF JUMP=2 THEN GOTO 3290
2880 IF JUMP=3 THEN GOTO 3680
2890 IF JUMP=4 THEN GOTO 4310
2900 '
2910 'Rotate spine plan around spine corner
2920 '
2930 LINE (SPINE(1,1),SPINE(1,2))-(SPINE(2,1),SPINE(2,2)),0
2940 LINE (SPINE(2,1),SPINE(2,2))-(SPINE(3,1),SPINE(3,2)),0
2950 LINE (SPINE(3,1),SPINE(3,2))-(SPINE(4,1),SPINE(4,2)),0
2960 LINE (SPINE(4,1),SPINE(4,2))-(SPINE(1,1),SPINE(1,2)),0
2970 FOR A=2 TO 4
2980     TSPINE(A,1)=SPINE(A,1)
2990     TSPINE(A,2)=SPINE(A,2)
3000 NEXT A
3010 FOR A=2 TO 4
3020      SPINE(A,1)=TSPINE(A,1)*COS(DEGREE)-TSPINE(A,2)*SIN(DEGREE)+
             SPINE(1,1)*(1-COS(DEGREE))+SPINE(1,2)*SIN(DEGREE)
3030      SPINE(A,2)=TSPINE(A,1)*SIN(DEGREE)+TSPINE(A,2)*COS(DEGREE)+
             SPINE(1,2)*(1-COS(DEGREE))-SPINE(1,1)*SIN(DEGREE)
3040 NEXT A
3041 FOR A=1 TO N
3042     XPLACE=INT((PLANPTS(TRACK(A,1),1,2)+(PLANPTS(TRACK(A,1),2,2)-
          PLANPTS(TRACK(A,1),1,2))/2)/8)
3043     YPLACE=INT((PLANPTS(TRACK(A,1),1,3)+(PLANPTS(TRACK(A,1),4,3)-
          PLANPTS(TRACK(A,1),1,3))/2)/8)
3044     LOCATE YPLACE,XPLACE
3045     IF A<10 THEN PRINT " "
3046     IF A>=10 THEN PRINT "  "
3047 NEXT A
3050 FOR A=1 TO N
3060 FLAG=1:I=1:COUNT=0
3070 WHILE FLAG
3080     I=PLANPTS(TRACK(A,1),I,1)
3090     LINE (PLANPTS(TRACK(A,1),PLANPTS(TRACK(A,1),I,1),2),PLANPTS(TRACK(A,1),
          PLANPTS(TRACK(A,1),I,1),3))-(PLANPTS(TRACK(A,1),PLANPTS(TRACK(A,1),I,4
          ),2),PLANPTS(TRACK(A,1),PLANPTS(TRACK(A,1),I,4),3)),0
3100     I=PLANPTS(TRACK(A,1),I,4)
3110     COUNT=COUNT+1
3120     IF COUNT>=SPACE(TRACK(A,1),11) THEN FLAG=0
3130 WEND
3140 NEXT A
3150 FOR A=1 TO N
3160     FOR B=1 TO SPACE(A,11)
3170      PLANPTS(A,B,2)=TPLANPTS(A,B,2)*COS(DEGREE)-TPLANPTS(A,B,3)*SIN(DEGREE)+
             SPINE(1,1)*(1-COS(DEGREE))+SPINE(1,2)*SIN(DEGREE)
3180      PLANPTS(A,B,3)=TPLANPTS(A,B,2)*SIN(DEGREE)+TPLANPTS(A,B,3)*COS(DEGREE)+
             SPINE(1,2)*(1-COS(DEGREE))-SPINE(1,1)*SIN(DEGREE)
3190     NEXT B
3200 NEXT A
3210 '
3220 'Draw rotated spine
```

```
3230 '
3240 LINE (SPINE(1,1),SPINE(1,2))-(SPINE(2,1),SPINE(2,2)),1
3250 LINE (SPINE(2,1),SPINE(2,2))-(SPINE(3,1),SPINE(3,2)),1
3260 LINE (SPINE(3,1),SPINE(3,2))-(SPINE(4,1),SPINE(4,2)),1
3270 LINE (SPINE(4,1),SPINE(4,2))-(SPINE(1,1),SPINE(1,2)),1
3280 GOTO 1210
3290 '
3300 'Rotate courtyard and activities
3310 '
3320 LINE (YARDPTS(1,1),YARDPTS(1,2))-(YARDPTS(2,1),YARDPTS(2,2)),0
3330 LINE (YARDPTS(2,1),YARDPTS(2,2))-(YARDPTS(3,1),YARDPTS(3,2)),0
3340 LINE (YARDPTS(3,1),YARDPTS(3,2))-(YARDPTS(4,1),YARDPTS(4,2)),0
3350 LINE (YARDPTS(4,1),YARDPTS(4,2))-(YARDPTS(1,1),YARDPTS(1,2)),0
3360 FOR A=2 TO 4
3370    TYARDPTS(A,1)=YARDPTS(A,1)
3380    TYARDPTS(A,2)=YARDPTS(A,2)
3390 NEXT A
3400 FOR A=2 TO 4
3410    YARDPTS(A,1)=TYARDPTS(A,1)*COS(DEGREE)-TYARDPTS(A,2)*SIN(DEGREE)+
           YARDPTS(1,1)*(1-COS(DEGREE))+YARDPTS(1,2)*SIN(DEGREE)
3420    YARDPTS(A,2)=TYARDPTS(A,1)*SIN(DEGREE)+TYARDPTS(A,2)*COS(DEGREE)+
           YARDPTS(1,2)*(1-COS(DEGREE))-YARDPTS(1,1)*SIN(DEGREE)
3430 NEXT A
3431 FOR A=1 TO N
3432    XPLACE=INT((PLANPTS(TRACK(A,1),1,2)+(PLANPTS(TRACK(A,1),2,2)-
         PLANPTS(TRACK(A,1),1,2))/2)/8)
3433    YPLACE=INT((PLANPTS(TRACK(A,1),1,3)+(PLANPTS(TRACK(A,1),4,3)-
         PLANPTS(TRACK(A,1),1,3))/2)/8)
3434    LOCATE YPLACE,XPLACE
3435    IF A<10 THEN PRINT " "
3436    IF A>=10 THEN PRINT "  "
3437 NEXT A
3440 FOR A=1 TO N
3450 FLAG=1:I=1:COUNT=0
3460 WHILE FLAG
3470    I=PLANPTS(TRACK(A,1),I,1)
3480    LINE (PLANPTS(TRACK(A,1),PLANPTS(TRACK(A,1),I,1),2),PLANPTS(TRACK(A,1),
         PLANPTS(TRACK(A,1),I,1),3))-(PLANPTS(TRACK(A,1),PLANPTS(TRACK(A,1),I,4
         ),2),PLANPTS(TRACK(A,1),PLANPTS(TRACK(A,1),I,4),3)),0
3490    I=PLANPTS(TRACK(A,1),I,4)
3500    COUNT=COUNT+1
3510    IF COUNT>=SPACE(TRACK(A,1),11) THEN FLAG=0
3520 WEND
3530 NEXT A
3540 FOR A=1 TO N
3550    FOR B=1 TO SPACE(A,11)
3560     PLANPTS(A,B,2)=TPLANPTS(A,B,2)*COS(DEGREE)-TPLANPTS(A,B,3)*SIN(DEGREE)+
            YARDPTS(1,1)*(1-COS(DEGREE))+YARDPTS(1,2)*SIN(DEGREE)
3570     PLANPTS(A,B,3)=TPLANPTS(A,B,2)*SIN(DEGREE)+TPLANPTS(A,B,3)*COS(DEGREE)+
            YARDPTS(1,2)*(1-COS(DEGREE))-YARDPTS(1,1)*SIN(DEGREE)
3580    NEXT B
3590 NEXT A
3600 '
3610 'Draw rotated courtyard
3620 '
3630 LINE (YARDPTS(1,1),YARDPTS(1,2))-(YARDPTS(2,1),YARDPTS(2,2)),1
3640 LINE (YARDPTS(2,1),YARDPTS(2,2))-(YARDPTS(3,1),YARDPTS(3,2)),1
3650 LINE (YARDPTS(3,1),YARDPTS(3,2))-(YARDPTS(4,1),YARDPTS(4,2)),1
3660 LINE (YARDPTS(4,1),YARDPTS(4,2))-(YARDPTS(1,1),YARDPTS(1,2)),1
3670 GOTO 1210
```

```
3680 '
3690 'Rotate pinwheel and activities
3700 '
3710 LINE (HALLPTS(12,1),HALLPTS(12,2))-(HALLPTS(1,1),HALLPTS(1,2)),0
3720 LINE (HALLPTS(1,1),HALLPTS(1,2))-(HALLPTS(5,1),HALLPTS(5,2)),0
3730 LINE (HALLPTS(5,1),HALLPTS(5,2))-(PINPTS(4,1),PINPTS(4,2)),0
3740 LINE (HALLPTS(9,1),HALLPTS(9,2))-(HALLPTS(2,1),HALLPTS(2,2)),0
3750 LINE (HALLPTS(2,1),HALLPTS(2,2))-(HALLPTS(6,1),HALLPTS(6,2)),0
3760 LINE (HALLPTS(6,1),HALLPTS(6,2))-(PINPTS(1,1),PINPTS(1,2)),0
3770 LINE (PINPTS(2,1),PINPTS(2,2))-(HALLPTS(7,1),HALLPTS(7,2)),0
3780 LINE (HALLPTS(7,1),HALLPTS(7,2))-(HALLPTS(3,1),HALLPTS(3,2)),0
3790 LINE (HALLPTS(3,1),HALLPTS(3,2))-(HALLPTS(10,1),HALLPTS(10,2)),0
3800 LINE (PINPTS(3,1),PINPTS(3,2))-(HALLPTS(8,1),HALLPTS(8,2)),0
3810 LINE (HALLPTS(8,1),HALLPTS(8,2))-(HALLPTS(4,1),HALLPTS(4,2)),0
3820 LINE (HALLPTS(4,1),HALLPTS(4,2))-(HALLPTS(11,1),HALLPTS(11,2)),0
3830 FOR A=2 TO 4
3840    TPINPTS(A,1)=PINPTS(A,1)
3850    TPINPTS(A,2)=PINPTS(A,2)
3860 NEXT A
3870 FOR A=2 TO 4
3880    PINPTS(A,1)=TPINPTS(A,1)*COS(DEGREE)-TPINPTS(A,2)*SIN(DEGREE)+
           PINPTS(1,1)*(1-COS(DEGREE))+PINPTS(1,2)*SIN(DEGREE)
3890    PINPTS(A,2)=TPINPTS(A,1)*SIN(DEGREE)+TPINPTS(A,2)*COS(DEGREE)+
           PINPTS(1,2)*(1-COS(DEGREE))-PINPTS(1,1)*SIN(DEGREE)
3900 NEXT A
3910 FOR A=1 TO 12
3920    THALLPTS(A,1)=HALLPTS(A,1)
3930    THALLPTS(A,2)=HALLPTS(A,2)
3940 FOR A=1 TO 12
3950 NEXT A
3960    HALLPTS(A,1)=THALLPTS(A,1)*COS(DEGREE)-THALLPTS(A,2)*SIN(DEGREE)+
           PINPTS(1,1)*(1-COS(DEGREE))+PINPTS(1,2)*SIN(DEGREE)
3970    HALLPTS(A,2)=THALLPTS(A,1)*SIN(DEGREE)+THALLPTS(A,2)*COS(DEGREE)+
           PINPTS(1,2)*(1-COS(DEGREE))-PINPTS(1,1)*SIN(DEGREE)
3980 NEXT A
3981 FOR A=1 TO N
3982    XPLACE=INT((PLANPTS(TRACK(A,1),1,2)+(PLANPTS(TRACK(A,1),2,2)-
         PLANPTS(TRACK(A,1),1,2))/2)/8)
3983    YPLACE=INT((PLANPTS(TRACK(A,1),1,3)+(PLANPTS(TRACK(A,1),4,3)-
         PLANPTS(TRACK(A,1),1,3))/2)/8)
3984    LOCATE YPLACE,XPLACE
3985    IF A<10 THEN PRINT " "
3986    IF A>=10 THEN PRINT "  "
3987 NEXT A
3990 FOR A=1 TO N
4000 FLAG=1:I=1:COUNT=0
4010 WHILE FLAG
4020    I=PLANPTS(TRACK(A,1),I,1)
4030    LINE (PLANPTS(TRACK(A,1),PLANPTS(TRACK(A,1),I,1),2),PLANPTS(TRACK(A,1),
         PLANPTS(TRACK(A,1),I,1),3))-(PLANPTS(TRACK(A,1),PLANPTS(TRACK(A,1),I,4
         ),2),PLANPTS(TRACK(A,1),PLANPTS(TRACK(A,1),I,4),3)),0
4040    I=PLANPTS(TRACK(A,1),I,4)
4050    COUNT=COUNT+1
4060    IF COUNT>=SPACE(TRACK(A,1),11) THEN FLAG=0
4070 WEND
4080 NEXT A
4090 FOR A=1 TO N
4100    FOR B=1 TO SPACE(A,11)
4110     PLANPTS(A,B,2)=TPLANPTS(A,B,2)*COS(DEGREE)-TPLANPTS(A,B,3)*SIN(DEGREE)+
            PINPTS(1,1)*(1-COS(DEGREE))+PINPTS(1,2)*SIN(DEGREE)
```

```
4120      PLANPTS(A,B,3)=TPLANPTS(A,B,2)*SIN(DEGREE)+TPLANPTS(A,B,3)*COS(DEGREE)+
             PINPTS(1,2)*(1-COS(DEGREE))-PINPTS(1,1)*SIN(DEGREE)
4130     NEXT B
4140 NEXT A
4150 '
4160 'Draw rotated pinwheel
4170 '
4180 LINE (HALLPTS(12,1),HALLPTS(12,2))-(HALLPTS(1,1),HALLPTS(1,2)),1
4190 LINE (HALLPTS(1,1),HALLPTS(1,2))-(HALLPTS(5,1),HALLPTS(5,2)),1
4200 LINE (HALLPTS(5,1),HALLPTS(5,2))-(PINPTS(4,1),PINPTS(4,2)),1
4210 LINE (HALLPTS(9,1),HALLPTS(9,2))-(HALLPTS(2,1),HALLPTS(2,2)),1
4220 LINE (HALLPTS(2,1),HALLPTS(2,2))-(HALLPTS(6,1),HALLPTS(6,2)),1
4230 LINE (HALLPTS(6,1),HALLPTS(6,2))-(PINPTS(1,1),PINPTS(1,2)),1
4240 LINE (PINPTS(2,1),PINPTS(2,2))-(HALLPTS(7,1),HALLPTS(7,2)),1
4250 LINE (HALLPTS(7,1),HALLPTS(7,2))-(HALLPTS(3,1),HALLPTS(3,2)),1
4260 LINE (HALLPTS(3,1),HALLPTS(3,2))-(HALLPTS(10,1),HALLPTS(10,2)),1
4270 LINE (PINPTS(3,1),PINPTS(3,2))-(HALLPTS(8,1),HALLPTS(8,2)),1
4280 LINE (HALLPTS(8,1),HALLPTS(8,2))-(HALLPTS(4,1),HALLPTS(4,2)),1
4290 LINE (HALLPTS(4,1),HALLPTS(4,2))-(HALLPTS(11,1),HALLPTS(11,2)),1
4300 GOTO 1210
4310 '
4320 'Rotate clustered activities
4330 '
4331 FOR A=1 TO N
4332     XPLACE=INT((PLANPTS(TRACK(A,1),1,2)+(PLANPTS(TRACK(A,1),2,2)-
          PLANPTS(TRACK(A,1),1,2))/2)/8)
4333     YPLACE=INT((PLANPTS(TRACK(A,1),1,3)+(PLANPTS(TRACK(A,1),4,3)-
          PLANPTS(TRACK(A,1),1,3))/2)/8)
4334     LOCATE YPLACE,XPLACE
4335     IF A<10 THEN PRINT " "
4336     IF A>=10 THEN PRINT "  "
4337 NEXT A
4340 FOR A=1 TO N
4350 FLAG=1:I=1:COUNT=0
4360 WHILE FLAG
4370     I=PLANPTS(TRACK(A,1),I,1)
4380     LINE (PLANPTS(TRACK(A,1),PLANPTS(TRACK(A,1),I,1),2),PLANPTS(TRACK(A,1),
          PLANPTS(TRACK(A,1),I,1),3))-(PLANPTS(TRACK(A,1),PLANPTS(TRACK(A,1),I,4
          ),2),PLANPTS(TRACK(A,1),PLANPTS(TRACK(A,1),I,4),3)),0
4390     I=PLANPTS(TRACK(A,1),I,4)
4400     COUNT=COUNT+1
4410     IF COUNT>=SPACE(TRACK(A,1),11) THEN FLAG=0
4420 WEND
4430 NEXT A
4440 FOR A=1 TO N
4450     FOR B=1 TO SPACE(A,11)
4460      PLANPTS(A,B,2)=TPLANPTS(A,B,2)*COS(DEGREE)-TPLANPTS(A,B,3)*SIN(DEGREE)+
             PLANPTS(TRACK(1,1),1,2)*(1-COS(DEGREE))
             +PLANPTS(TRACK(1,1),1,3)*SIN(DEGREE)
4470      PLANPTS(A,B,3)=TPLANPTS(A,B,2)*SIN(DEGREE)+TPLANPTS(A,B,3)*COS(DEGREE)+
             PLANPTS(TRACK(1,1),1,3)*(1-COS(DEGREE))
             -PLANPTS(TRACK(1,1),1,2)*SIN(DEGREE)
4480     NEXT B
4490 NEXT A
4500 GOTO 1210
4510 '
4520 'Modify or redraw activities
4530 '
4540 LINE (2,165)-(75,150),0,B:LINE (82,165)-(150,150),0,B:
4550 LOCATE 20,1:PRINT SPC(39)
```

```
4560 LINE (2,185)-(75,170),0,B:LINE (82,185)-(150,170),0,B:
     LINE(160,185)-(240,170),0,B
4570 LOCATE 23,1:PRINT "INDICATE NEW ACTIVITY LOCATION   "
4580 FOR A=1 TO 500:NEXT A
4590 PEN ON
4600 IF PEN(3)=0 THEN GOTO 4590
4610 X1=PEN(4):Y1=PEN(5)
4620 PSET (X1,Y1)
4630 LOCATE 23,1:PRINT "CORRECT POINT?     YES   NO     "
4640 LINE (150,185)-(175,170),1,B:LINE (195,185)-(220,170),1,B
4650 PEN ON
4660 IF PEN(3)=0 THEN GOTO 4650
4670 IF PEN(8)>21 AND PEN(9)>24 THEN PSET (X1,Y1),0:GOTO 4580 ELSE GOTO 4680
4680 PSET (X1,Y1),0
4690 FOR A=1 TO N
4700    IF X1>=PLANPTS(A,1,2) AND X1<=PLANPTS(A,2,2) AND Y1>=PLANPTS(A,1,3)
        AND Y1<=PLANPTS(A,4,3) THEN ACTIVITY=A:GOTO 4730 ELSE GOTO 4710
4710 NEXT A
4720 STOP
4730 LOCATE 1,38:PRINT "*"
4740 FOR B=1 TO 500:NEXT B
4750 PEN ON
4760 IF PEN(3)=0 THEN GOTO 4750
4770 X2=PEN(4):Y2=PEN(5)
4780 PSET (X2,Y2)
4781 FOR B=1 TO 500:NEXT B
4790 PEN ON
4800 IF PEN(3)=0 THEN GOTO 4790
4810 IF PEN(8)>21 AND PEN(9)>24 THEN PSET (X2,Y2),0:GOTO 4740 ELSE GOTO 4820
4820 PSET (X2,Y2),0
4830 LINE (150,185)-(175,170),0,B:LINE (195,185)-(220,170),0,B
4840 XPLACE=INT((PLANPTS(ACTIVITY,1,2)+(PLANPTS(ACTIVITY,2,2)-
     PLANPTS(ACTIVITY,1,2))/2)/8)
4850 YPLACE=INT((PLANPTS(ACTIVITY,1,3)+(PLANPTS(ACTIVITY,4,3)-
     PLANPTS(ACTIVITY,1,3))/2)/8)
4860 LOCATE YPLACE,XPLACE
4870 IF ACTIVITY<10 THEN PRINT " "
4880 IF ACTIVITY>=10 THEN PRINT "  "
4890 FLAG=1:I=1:COUNT=0
4900 WHILE FLAG
4910    I=PLANPTS(ACTIVITY,I,1)
4920    LINE (PLANPTS(ACTIVITY,PLANPTS(ACTIVITY,I,1),2),PLANPTS(ACTIVITY,
        PLANPTS(ACTIVITY,I,1),3))-(PLANPTS(ACTIVITY,PLANPTS(ACTIVITY,I,4),2)
        ,PLANPTS(ACTIVITY,PLANPTS(ACTIVITY,I,4),3)),0
4930    I=PLANPTS(ACTIVITY,I,4)
4940    COUNT=COUNT+1
4950    IF COUNT>=SPACE(ACTIVITY,11) THEN FLAG=0
4960 WEND
4970 IF JUMP=1 THEN GOTO 5010
4980 IF JUMP=2 THEN GOTO 5140
4990 IF JUMP=3 THEN GOTO 5190
5000 IF JUMP=4 THEN GOTO 5310
5010 '
5020 'Line modification
5030 '
5040 WSPINE=SPINE(4,2)-SPINE(1,2)
5050 XDIST=X2-PLANPTS(ACTIVITY,1,2)
5060 IF PLANPTS(ACTIVITY,4,3)=SPINE(1,2) AND Y2<=SPINE(1,2)+WSPINE/2
     THEN YDIST=0
5070 IF PLANPTS(ACTIVITY,4,3)=SPINE(1,2) AND Y2> SPINE(1,2)+WSPINE/2 THEN
```

```
      YDIST=PLANPTS(ACTIVITY,3,3)-PLANPTS(ACTIVITY,2,3)+WSPINE:
      SPACE(ACTIVITY,6)=3 ELSE GOTO 5080
5080 IF PLANPTS(ACTIVITY,1,3)=SPINE(1,2)+WSPINE AND Y2<=SPINE(1,2)+WSPINE/2 THEN
      YDIST=PLANPTS(ACTIVITY,2,3)-PLANPTS(ACTIVITY,3,3)-WSPINE:
      SPACE(ACTIVITY,6)=1 ELSE GOTO 5090
5090 IF PLANPTS(ACTIVITY,1,3)=SPINE(1,2)+WSPINE AND Y2> SPINE(1,2)+WSPINE/2
      THEN YDIST=0
5100 GOTO 5360
5110 '
5120 'Courtyard modification
5130 '
5140 IF X2>=YARDPTS(1,1)-8 AND X2<=YARDPTS(1,1)+8 THEN
      XDIST=YARDPTS(1,1)-PLANPTS(ACTIVITY,2,2):YDIST=Y2-PLANPTS(ACTIVITY,2,3):
      SPACE(ACTIVITY,6)=4:GOTO 5180 ELSE GOTO 5150
5150 IF X2>=YARDPTS(2,1)-8 AND X2<=YARDPTS(2,1)+8 THEN
      XDIST=YARDPTS(2,1)-PLANPTS(ACTIVITY,1,2):YDIST=Y2-PLANPTS(ACTIVITY,1,3):
      SPACE(ACTIVITY,6)=2:GOTO 5180 ELSE GOTO 5160
5160 IF Y2>=YARDPTS(1,2)-8 AND Y2<=YARDPTS(1,2)+8 THEN
      XDIST=X2-PLANPTS(ACTIVITY,4,2):YDIST=YARDPTS(1,2)-PLANPTS(ACTIVITY,4,3):
      SPACE(ACTIVITY,6)=1:GOTO 5180 ELSE GOTO 5170
5170 IF Y2>=YARDPTS(4,2)-8 AND Y2<=YARDPTS(4,2)+8 THEN
      XDIST=X2-PLANPTS(ACTIVITY,1,2):YDIST=YARDPTS(4,2)-PLANPTS(ACTIVITY,1,3):
      SPACE(ACTIVITY,6)=3:GOTO 5180 ELSE GOTO 4560
5180 GOTO 5360
5190 '
5200 'Radial modification
5210 '
5220 IF Y2>=PINPTS(1,2)-3 AND Y2<=PINPTS(1,2)+5 AND X2>PINPTS(1,1) THEN
      XDIST=X2-PLANPTS(ACTIVITY,1,2):YDIST=PINPTS(1,2)-PLANPTS(ACTIVITY,1,3):
      SPACE(ACTIVITY,6)=1:GOTO 5300 ELSE GOTO 5230
5230 IF Y2>=PINPTS(4,2)-5 AND Y2<=PINPTS(4,2)+3 AND X2<PINPTS(3,1) THEN
      XDIST=X2-PLANPTS(ACTIVITY,4,2):YDIST=PINPTS(4,2)-PLANPTS(ACTIVITY,4,3):
      SPACE(ACTIVITY,6)=3:GOTO 5300 ELSE GOTO 5240
5240 IF X2>=PINPTS(2,1)-5 AND X2<=PINPTS(2,1)+3 AND Y2>PINPTS(2,2)+5 THEN
      XDIST=PINPTS(2,1)-PLANPTS(ACTIVITY,2,2):YDIST=Y2-PLANPTS(ACTIVITY,2,3):
      SPACE(ACTIVITY,6)=2:GOTO 5300 ELSE GOTO 5250
5250 IF X2>=PINPTS(1,1)-3 AND X2<=PINPTS(1,1)+5 AND Y2<PINPTS(4,2)-5 THEN
      XDIST=PINPTS(1,1)-PLANPTS(ACTIVITY,1,2):YDIST=Y2-PLANPTS(ACTIVITY,1,3):
      SPACE(ACTIVITY,6)=4:GOTO 5300 ELSE GOTO 5260
5260 IF Y2>=HALLPTS(2,2)-5 AND Y2<=HALLPTS(2,2)+2 AND X2>PINPTS(1,1) THEN
      XDIST=X2-PLANPTS(ACTIVITY,4,2):YDIST=HALLPTS(2,2)-PLANPTS(ACTIVITY,4,3):
      SPACE(ACTIVITY,6)=1:GOTO 5300 ELSE GOTO 5270
5270 IF Y2>=HALLPTS(4,2)-2 AND Y2<=HALLPTS(4,2)+5 AND X2<PINPTS(3,1) THEN
      XDIST=X2-PLANPTS(ACTIVITY,1,2):YDIST=HALLPTS(4,2)-PLANPTS(ACTIVITY,1,3):
      SPACE(ACTIVITY,6)=3:GOTO 5300 ELSE GOTO 5280
5280 IF X2>=HALLPTS(3,1)-2 AND X2<=HALLPTS(3,1)+5 AND Y2>PINPTS(2,2)+5 THEN
      XDIST=HALLPTS(3,1)-PLANPTS(ACTIVITY,1,2):YDIST=Y2-PLANPTS(ACTIVITY,1,3):
      SPACE(ACTIVITY,6)=2:GOTO 5300 ELSE GOTO 5290
5290 IF X2>=HALLPTS(1,1)-5 AND X2<=HALLPTS(1,1)+2 AND Y2< PINPTS(4,2)-5 THEN
      XDIST=HALLPTS(1,1)-PLANPTS(ACTIVITY,2,2):YDIST=Y2-PLANPTS(ACTIVITY,2,3):
      SPACE(ACTIVITY,6)=4:GOTO 5300 ELSE GOTO 4560
5300 GOTO 5360
5310 '
5320 'Cluster modification
5330 '
5340 XDIST=X2-PLANPTS(ACTIVITY,1,2):YDIST=Y2-PLANPTS(ACTIVITY,1,3)
5350 GOTO 5360
5360 FOR A=1 TO SPACE(ACTIVITY,11)
5370    PLANPTS(ACTIVITY,A,2)=PLANPTS(ACTIVITY,A,2)+XDIST
5380    PLANPTS(ACTIVITY,A,3)=PLANPTS(ACTIVITY,A,3)+YDIST
```

```
5390 NEXT A
5400 FLAG=1:I=1:COUNT=0
5410 WHILE FLAG
5420    I=PLANPTS(ACTIVITY,I,1)
5430    LINE (PLANPTS(ACTIVITY,PLANPTS(ACTIVITY,I,1),2),PLANPTS(ACTIVITY,
        PLANPTS(ACTIVITY,I,1),3))-(PLANPTS(ACTIVITY,PLANPTS(ACTIVITY,I,4),2)
        ,PLANPTS(ACTIVITY,PLANPTS(ACTIVITY,I,4),3)),1
5440    I=PLANPTS(ACTIVITY,I,4)
5450    COUNT=COUNT+1
5460    IF COUNT>=SPACE(ACTIVITY,11) THEN FLAG=0
5470 WEND
5480 XPLACE=INT((PLANPTS(ACTIVITY,1,2)+(PLANPTS(ACTIVITY,2,2)-
     PLANPTS(ACTIVITY,1,2))/2)/8)
5490 YPLACE=INT((PLANPTS(ACTIVITY,1,3)+(PLANPTS(ACTIVITY,4,3)-
       PLANPTS(ACTIVITY,1,3))/2)/8)
5500 LOCATE YPLACE,XPLACE
5510 IF ACTIVITY<10 THEN PRINT USING "#";ACTIVITY
5520 IF ACTIVITY>=10 THEN PRINT USING "##";ACTIVITY
5530 GOTO 1520
5540 '
5550 'Redraw activities
5560 '
5570 LINE (2,185)-(75,170),0,B:LINE (82,185)-(150,170),0,B:
     LINE(160,185)-(240,170),0,B
5580 LOCATE 23,1:PRINT SPC(39)
5590 LOCATE 20,1:PRINT SPC(39)
5600 LINE (2,165)-(75,150),0,B:LINE (82,165)-(150,150),0,B:
5610 FOR A=1 TO N
5620 FLAG=1:I=1:COUNT=0
5630 WHILE FLAG
5640    I=PLANPTS(TRACK(A,1),I,1)
5650    LINE (PLANPTS(TRACK(A,1),PLANPTS(TRACK(A,1),I,1),2),PLANPTS(TRACK(A,1),
         PLANPTS(TRACK(A,1),I,1),3))-(PLANPTS(TRACK(A,1),PLANPTS(TRACK(A,1),I,4
         ),2),PLANPTS(TRACK(A,1),PLANPTS(TRACK(A,1),I,4),3)),1
5660    I=PLANPTS(TRACK(A,1),I,4)
5670    COUNT=COUNT+1
5680    IF COUNT>=SPACE(TRACK(A,1),11) THEN FLAG=0
5690 WEND
5700 NEXT A
5710 IF JUMP=1 THEN GOTO 5750
5720 IF JUMP=2 THEN GOTO 5770
5730 IF JUMP=3 THEN GOTO 5790
5740 IF JUMP=4 THEN GOTO 4510
5750 LINE (SPINE(1,1),SPINE(1,2))-(SPINE(3,1),SPINE(3,2)),1,B
5760 GOTO 1520
5770 LINE (YARDPTS(1,1),YARDPTS(1,2))-(YARDPTS(3,1),YARDPTS(3,2)),1,B
5780 GOTO 1520
5790 LINE (PINPTS(1,1),PINPTS(1,2))-(PINPTS(3,1),PINPTS(3,2)),1,B
5800 LINE (HALLPTS(1,1),HALLPTS(1,2))-(PINPTS(4,1),PINPTS(4,2)),1,B
5810 LINE (HALLPTS(2,1),HALLPTS(2,2))-(PINPTS(1,1),PINPTS(1,2)),1,B
5820 LINE (HALLPTS(3,1),HALLPTS(3,2))-(PINPTS(2,1),PINPTS(2,2)),1,B
5830 LINE (HALLPTS(4,1),HALLPTS(4,2))-(PINPTS(3,1),PINPTS(3,2)),1,B
5840 GOTO 1520
5850 '
5860 'Store plan and site points in temporary files
5870 '
5880 FILENM1$="B:PLAN":FILENM2$="B:SITE"
5890 OPEN FILENM1$ FOR OUTPUT AS #1
5900 PRINT#1,JUMP
5910 PRINT#1,N
```

```
5920 IF JUMP=1 THEN GOTO 5960
5930 IF JUMP=2 THEN GOTO 6010
5940 IF JUMP=3 THEN GOTO 6060
5950 IF JUMP=4 THEN GOTO 6140
5960 FOR A=1 TO 4
5970    PRINT#1,SPINE(A,1)
5980    PRINT#1,SPINE(A,2)
5990 NEXT A
6000 GOTO 6140
6010 FOR A=1 TO 4
6020    PRINT#1,YARDPTS(A,1)
6030    PRINT#1,YARDPTS(A,2)
6040 NEXT A
6050 GOTO 6140
6060 FOR A=1 TO 4
6070    PRINT#1,PINPTS(A,1)
6080    PRINT#1,PINPTS(A,2)
6090 NEXT A
6100 FOR A=1 TO 12
6110    PRINT#1,HALLPTS(A,1)
6120    PRINT#1,HALLPTS(A,2)
6130 NEXT A
6140 FOR A=1 TO N
6150    PRINT#1,SPACE(A,4)
6160    PRINT#1,SPACE(A,11)
6170 NEXT A
6180 FOR A=1 TO N
6190    PRINT#1,TRACK(A,1)
6200 NEXT A
6210 FOR A=1 TO N
6220    FOR B=1 TO SPACE(A,11)
6230      PRINT#1,PLANPTS(A,B,1)
6240      PRINT#1,PLANPTS(A,B,2)
6250      PRINT#1,PLANPTS(A,B,3)
6260      PRINT#1,PLANPTS(A,B,4)
6270    NEXT B
6280 NEXT A
6290 FOR A=1 TO N
6300    PRINT#1,TITLE$(A,A)
6310 NEXT A
6320 CLOSE#1
6330 OPEN FILENM2$ FOR OUTPUT AS #2
6340 PRINT#2,NUMSITEPOINTS
6350 PRINT#2,NUMEASEPOINTS
6360 PRINT#2,NUMTREES
6370 FOR A=1 TO NUMSITEPOINTS
6380    PRINT#2,SITE(A,1)
6390    PRINT#2,SITE(A,2)
6400 NEXT A
6410 FOR A=1 TO NUMEASEPOINTS
6420    PRINT#2,EASE(A,1)
6430    PRINT#2,EASE(A,2)
6440 NEXT A
6450 FOR A=1 TO NUMTREES
6460    PRINT#2,TREE(A,1)
6470    PRINT#2,TREE(A,2)
6480    PRINT#2,TREE(A,3)
6490 NEXT A
6500 CLOSE#2
6510 '
```

```
6520 'Allow another circulation design to be developed
6530 '
6540 ERASE TSPINE,TYARDPTS,TPINPTS,THALLPTS,TPLANPTS
6550 LOCATE 20,1:PRINT SPC(30)
6560 LINE (2,165)-(75,150),0,B:LINE (82,165)-(150,150),0,B:
6570 LINE (2,185)-(75,170),0,B:LINE (82,185)-(150,170),0,B:
     LINE(160,185)-(240,170),0,B
6580 LOCATE 23,1:PRINT "DRAW ELEVATIONS?   YES   NO     "
6590 LINE (150,185)-(175,170),1,B:LINE (195,185)-(220,170),1,B
6600 PEN ON
6610 IF PEN(3)=0 THEN GOTO 6600
6620 IF PEN(8)>21 AND PEN(9)<23 THEN GOTO 6860
6630 LOCATE 23,1:PRINT "SAVE DRAWING?      YES   NO     "
6640 LINE (150,185)-(175,170),1,B:LINE (195,185)-(220,170),1,B
6650 PEN ON
6660 IF PEN(3)=0 THEN GOTO 6650
6670 IF PEN(8)>21 AND PEN(9)<23 THEN JUMP$="PICTURE":GOTO 6850 ELSE GOTO 6680
6680 LOCATE 23,1:PRINT "MODIFY ANOTHER?    YES   NO     "
6690 LINE (150,185)-(175,170),1,B:LINE (195,185)-(220,170),1,B
6700 PEN ON
6710 IF PEN(3)=0 THEN GOTO 6700
6720 IF PEN(8)>21 AND PEN(9)<23 THEN GOTO 50
6730 LOCATE 23,1:PRINT "SITE PLAN?         YES   NO     "
6740 LINE (150,185)-(175,170),1,B:LINE (195,185)-(220,170),1,B
6750 PEN ON
6760 IF PEN(3)=0 THEN GOTO 6750
6770 IF PEN(8)>21 AND PEN(9)<23 THEN GOTO 6840
6780 LOCATE 23,1:PRINT "START AGAIN?       YES   NO     "
6790 LINE (150,185)-(175,170),1,B:LINE (195,185)-(220,170),1,B
6800 PEN ON
6810 IF PEN(3)=0 THEN GOTO 6800
6820 IF PEN(8)>21 AND PEN(9)>24 THEN STOP
6830 CHAIN MERGE "CREATE",10,DELETE 50-6820
6840 CHAIN MERGE "SITE",10,ALL,DELETE 50-6830
6850 CHAIN MERGE "PLNFILE",10,ALL,DELETE 50-6840
6860 CHAIN MERGE "ELEVTION",10,ALL,DELETE 50-6850
```

```
10 '*************************************************************************
20 'PLNFILE Subroutine
30 '*************************************************************************
40 'This routine stores the plan designs into File 1 and the site plans
50 'into File 2.  Control returns to the routine from which called.
60 '
70 CLS
80 LOCATE 2,13:PRINT "PLNFILE Subroutine"
90 LOCATE 23,1:PRINT SPC(39)
100 IF JUMP$="SITE1" THEN GOTO 570
110 LOCATE 23,1:INPUT "ENTER FILE NAME FOR PLAN: ",FILENM$
120 OPEN FILENM$ FOR OUTPUT AS #1
130 PRINT#1,JUMP
140 PRINT#1,N
150 IF JUMP=1 THEN GOTO 190
160 IF JUMP=2 THEN GOTO 240
170 IF JUMP=3 THEN GOTO 290
180 IF JUMP=4 THEN GOTO 370
190 FOR A=1 TO 4
200     PRINT#1,SPINE(A,1)
210     PRINT#1,SPINE(A,2)
220 NEXT A
230 GOTO 370
240 FOR A=1 TO 4
250     PRINT#1,YARDPTS(A,1)
260     PRINT#1,YARDPTS(A,2)
270 NEXT A
280 GOTO 370
290 FOR A=1 TO 4
300     PRINT#1,PINPTS(A,1)
310     PRINT#1,PINPTS(A,2)
320 NEXT A
330 FOR A=1 TO 12
340     PRINT#1,HALLPTS(A,1)
350     PRINT#1,HALLPTS(A,2)
360 NEXT A
370 FOR A=1 TO N
380     PRINT#1,SPACE(A,4)
390     PRINT#1,SPACE(A,11)
400 NEXT A
410 FOR A=1 TO N
420     PRINT#1,TRACK(A,1)
430 NEXT A
440 FOR A=1 TO N
450      FOR B=1 TO SPACE(A,11)
460        PRINT#1,PLANPTS(A,B,1)
470        PRINT#1,PLANPTS(A,B,2)
480        PRINT#1,PLANPTS(A,B,3)
490        PRINT#1,PLANPTS(A,B,4)
500     NEXT B
510 NEXT A
520 FOR A=1 TO N
530     PRINT#1,TITLE$(A,A)
540 NEXT A
550 CLOSE#1
560 IF JUMP$<>"SITE" THEN GOTO 770
570 LOCATE 23,1:PRINT SPC(39)
580 LOCATE 23,1:INPUT "ENTER FILE NAME FOR SITE: ",FILENM2$
590 OPEN FILENM2$ FOR OUTPUT AS #2
600 PRINT#2,NUMSITEPOINTS
```

```
610 PRINT#2,NUMEASEPOINTS
620 PRINT#2,NUMTREES
630 FOR A=1 TO NUMSITEPOINTS
640     PRINT#2,SITE(A,1)
650     PRINT#2,SITE(A,2)
660 NEXT A
670 FOR A=1 TO NUMEASEPOINTS
680     PRINT#2,EASE(A,1)
690     PRINT#2,EASE(A,2)
700 NEXT A
710 FOR A=1 TO NUMTREES
720     PRINT#2,TREE(A,1)
730     PRINT#2,TREE(A,2)
740     PRINT#2,TREE(A,3)
750 NEXT A
760 CLOSE #2
770 IF JUMP$="SPINE" THEN GOTO 840
780 IF JUMP$="COURTYARD" THEN GOTO 850
790 IF JUMP$="RADIAL" THEN GOTO 860
800 IF JUMP$="CLUSTER" THEN GOTO 870
810 IF JUMP$="SITE" THEN GOTO 880
820 IF JUMP$="SITE1" THEN GOTO 890
830 IF JUMP$="PICTURE" THEN GOTO 900
840 CHAIN MERGE "LINE",5980,ALL,DELETE 10-830
850 CHAIN MERGE "POINT",5450,ALL,DELETE 10-840
860 CHAIN MERGE "RADIAL",8860,ALL,DELETE 10-850
870 CHAIN MERGE "CLUSTER",3010,ALL,DELETE 10-860
880 CHAIN MERGE "SITE",6110,ALL,DELETE 10-870
890 CHAIN MERGE "SITE",1920,ALL,DELETE 10-880
900 CHAIN MERGE "PICTURE",5560,ALL,DELETE 10-890
```

APPENDIX B

SPINE PLACEMENT SCHEME GUIDELINES

Scheme Number	Wall Assignment Placed	Wall Assignment Candidate	Guidelines
Note:	CREATE checks at each stage of the guidelines, for all schemes, to insure the candidate activity does not extend into the reserved corner space.		
1	North OR North	North None	a. Check if space available adjacent to east side of placed activity and place candidate if possible. b. If not placed, check to west of placed activity to insure candidate activity will not extend into reserved corner space. c. Check if space available adjacent to west side of placed activity and place candidate if possible. d. If not, check for available space to the east of the placed activity and place if possible. e. If not placed, check for space available to the west of the placed activity and place if possible. f. If not placed, display default screen and allow user to place.
2	South OR South	South None	a. Check if space available adjacent to west side of placed activity and place candidate if possible. b. If not placed, check to the east of the placed activity to insure candidate activity will not extend into the reserved corner space.

Scheme Number	Wall Assignment Placed	Candidate	Guidelines
			c. Check if space available adjacent to the east side of the placed activity and place candidate if possible.
			d. If not, check west side of placed activity for available space and place if possible.
			e. If not placed, check east side of placed activity for available space and place if possible.
			f. If not, display the default screen and allow user to place.
3	North	South	a. Check if candidate activity may be placed on south side of spine directly across from placed activity.
			b. If not follow guidelines described for Scheme 2.
4	South	North	a. Check if candidate activity may be placed on north side of spine directly across from placed activity.
			b. If not placed, follow guidelines described for Scheme 1.
5	North	East	a. Check if space available for placement on north side of spine on the east side at the midpoint and place if possible.
			b. If not placed, check if space available for placement on south side of spine at the midpoint and place if possible.
			c. If not, check space on north side of spine, east of the midpoint for space available for placement
			d. If not, check for available space on south side of spine, east of the midpoint and place if possible.

Scheme Number	Wall Assignment Placed	Wall Assignment Candidate	Guidelines
			e. If not placed, display default screen and allow user to place.
6	South	East	a. Check if space available for placement on south side of spine, on the east side at the midpoint. Place if possible.
			b. If not placed, check for space available for placement on north side of spine, on the east side at the midpoint. Place if possible.
			c. If not, check for space on south side of spine, east of the midpoint for available space and place if possible.
			d. If not placed, check for space on the south side of the spine, east of the midpoint and place if possible.
			e. If not placed, display default screen.
7	East	North	a. Check if space available for activity placement at midpoint on the north side of the spine. Place if possible.
			b. If not, follow the guidelines outlined for Scheme 1.
8	East	South	a. Check if space available for activity placement at midpoint on the south side of the spine. Place if possible.
			b. If not placed, follow the guidelines outlined for Scheme 2
9	North	West	a. Check if space available on the north side, adjacent to and west of the spine midpoint. Place if possible.
			b. If not placed, check for space available on south side of spine, adjacent to and west of the spine midpoint. Place if possible

Scheme Number	Wall Assignment Placed	Wall Assignment Candidate	Guidelines
			c. If not, check for space available on the north side of the spine, west of the midpoint and place if possible.
			d. If not placed, check for space available on the south side of the spine, west of the midpoint and place if possible.
			e. If not placed, display the default screen.
10	South	West	a. Check if space available on the south side, adjacent to and west of the spine midpoint. Place if possible.
			b. If not placed, check for space available on the north side, adjacent to and west of the midpoint. Place if possible.
			c. If not check for space available on the south side of the spine, west of the midpoint and place if possible.
			d. If not placed, check for available space on the north side. west of the midpoint and place if possible.
			e. If not placed, display the default screen.
11	West	North	a. Check if space available at midpoint on north side of spine and place if possible.
			b. If not, follow the guidelines described for Scheme 1.
12	West	South	a. Check if space available at midpoint on south side of spine and place if possible
			b. If not, follow the guidelines described for Scheme 2

Scheme Number	Wall Assignment Placed	Candidate	Guidelines
13	East	West	a. Check if space available at midpoint on north side of spine and place if possible.
			b. If not placed check if space available at midpoint on south side of spine and place if possible.
			c. If not, check the south side of the spine, west of the midpoint, for available space and place if possible.
			d. If not placed, check the north side of the spine, west of the midpoint and place if possible.
			e. If not, display default screen.
14	West	East	a. Check if space available adjacent to and east of the midpoint on north side of spine and place if possible.
			b. If not placed, check for available space adjacent to and east of the midpoint on south side of the spine and place if possible.
			c. If not placed, check north side, east of midpoint for space and place if possible.
			d. If not, check south side of spine, east of midpoint and place if possible.
			e. If not placed, display the default screen.
15	East	East	a. Check for space available, east of the midpoint on the north side of the spine and place if possible.
	OR		
	East	None	
			b. If not, check for available space east of the midpoint on the south side of the spine and place if possible.
			c. If not placed. display the default screen.

Scheme Number	Wall Assignment Placed	Wall Assignment Candidate	Guidelines
16	West OR West	West None	a. Check for space available west of the spine midpoint on the north side of the spine. Place if possible. b. If not, check for space available, west of the spine midpoint on the south side of the spine. Place if possible. c. If not placed, display default screen.

APPENDIX C

COURTYARD PLACEMENT SCHEME GUIDELINES

Scheme Number	Wall Assignment Placed	Wall Assignment Candidate	Guidelines
NOTE:	CREATE checks at each stage of the guidelines, for all schemes, to insure the candidate activity does not extend into the reserved corner space.		
1	North OR North	North None	a. Check if space available at initial north corner position and place if possible. b. If not placed, check for space available adjacent to and east of the placed activity and place if possible. c. Check for space available on the north side of the courtyard and place if possible. d. If not placed display the default screen and allow the user to place the candidate activity.
2	South OR South	South None	a. Check if space available at initial south corner position and place if possible. b. If not placed, check for available space directly west of the south placed activity and place if possible. c. If not placed, check for space available on the south side of the courtyard and place if possible. d. If not placed, display the default screen.
3	North	South	a. Check if space available at the initial south corner position and place if possible. b. If not placed, determine strongest relationship between candidate activity and a south placed activity.

Scheme Number	Wall Assignment Placed	Candidate	Guidelines
			c. If not placed, check for available space directly west of this placed activity.
			d. If not, check for available space on the south side of the courtyard.
			e. If not placed, display the default screen.
4	South	North	a. Check if space available at initial north corner position and place if possible.
			b. If not placed, determine strongest relationship between the candidate activity and a north placed activity.
			c. Check if space available directly east of this placed activity and place if possible.
			d. If not, check for space available on the north side of the courtyard and place if possible.
			e. If not placed, display the default screen
5	East	East	a. Check if space available at initial east corner position and place if possible.
	OR		
	East	None	b. If not placed, check for space available directly south of the east placed activity and place if possible.
			c. If not, check for available space on east side of the courtyard and place if possible.
			d. If not, display the default screen.
6	West	West	a. Check if space available at initial west corner position and place if possible.
	OR		
	West	None	b. If not placed, check for space available directly north of the placed activity and place if possible.
			c. If not, check for available space on the west side of the courtyard and place if possible.
			d. If not placed, display the default screen.
7	East	West	a. Check if space available at the initial west corner position and place if possible.

Scheme Number	Wall Assignment Placed	Candidate	Guidelines
			b. If not, determine strongest relationship between candidate activity and west placed activity.
			c. Check if space available directly north of this placed activity and place if possible.
			d. If not placed. check for space available on the west side of the courtyard and place if possible.
			e. If not, display default screen.
8	West	East	a. Check if space available at the initial east corner position and place if possible.
			b. If not placed, determine the strongest relationship between candidate activity and east placed activity.
			c. Check if space available directly south of this placed activity and place if possible.
			d. If not, check for space available on east side of courtyard and place if possible.
			e. If not, display default screen
9	West	North	a. Check if space available at the initial north corner position and place if possible.
			b. If not placed, determine the strongest relationship between candidate activity and a north placed activity.
			c. Check for space available directly east of this placed activity and place if possible.
			d. If not placed, check for available space on north side of courtyard and place if possible.
			e. If not, display the default screen.
10	East	North	a. Follow guidelines outlined for Scheme 9.

Scheme Number	Wall Assignment Placed	Candidate	Guidelines
11	West	South	a. Check if space available at the initial south corner position and place if possible. b. If not, determine the strongest relationship between candidate activity and south placed activity. c. Check if space available directly west of this placed activity and place if possible. d. If not, check for available space on south side of courtyard and place if possible. e. If not placed, display default screen.
12	East	South	a. Follow guidelines describing Scheme 11.
13	South	West	a. Check if space available at the initial west corner position and place if possible. b. If not, determine strongest relationship between candidate activity and west placed activity. c. Check if space available directly north of this placed activity and place if possible. d. If not placed, check for space available on the west side of the courtyard and place if possible. e. If not, display default screen.
14	North	West	a. Check if space available at the initial west corner position and place if possible. b. If not, determine strongest relationship between candidate activity and west placed activity. c. Check if space available directly north of this placed activity and place if possible. d. If not placed, check for space available on the west side of the courtyard and place if possible.

Scheme Number	Wall Assignment Placed	Wall Assignment Candidate	Guidelines
			e. If not, display default screen.
15	North	East	a. Check if space available at the initial east corner position and place if possible.
			b. If not placed, determine the strongest relationship between candidate activity and east placed activity.
			c. Check if space available directly south of this placed activity and place if possible.
			d. If not, check for space available on east side of courtyard and place if possible.
			e. If not, display default screen.
16	South	East	a. Check if space available at the initial east corner position and place if possible.
			b. If not placed, determine the strongest relationship between candidate activity and east placed activity.
			c. Check if space available directly south of this placed activity and place if possible.
			d. If not, check for space available on east side of courtyard and place if possible.
			e. If not, display default screen.

APPENDIX D

RADIAL PLACEMENT SCHEME GUIDELINES

Scheme Number	Wall Assignment Placed	Wall Assignment Candidate	Guidelines
NOTE:	CREATE checks at each stage of the guidelines, for all schemes, to insure the candidate activity does not extend into the reserved corner space.		
1	North OR North	North None	a. Check if space available inside the north side courtyard area and place if possible. b. If not, check if space available on north side of pinwheel and place if possible. c. If not, check if space available along the north wing wall and place if possible. d. If not placed check for space on the north side of west wing wall and place if possible. e. If not, display the default screen and allow the user to place activity. (Refer to Figure 17 for graphical representation.)
2	South OR South	South None	a. Check if space available inside the south side of the courtyard and place if possible. b. If not placed, check for space on the south side of the pinwheel and place if possible. c. If not, check for available space on the south wing wall and place if possible. d. If not placed, check for space along the south side of the northeast wing wall and place if possible. e. If not, display default screen.

Scheme Number	Wall Assignment Placed	Wall Assignment Candidate	Guidelines
3	North	South	a. Check if space available on south side of the east wing wall and place if possible. b. If not placed, follow the procedures outlined in Scheme 2, steps a through c, and place if possible. c. If not placed, display the default screen.
4	South	North	a. Check if space available on the north side of the west wing wall and place if possible. b. If not placed, follow the procedures outlined in Scheme 1, steps a through c, and place if possible. c. If not placed, display default screen.
5	East OR East	East None	a. Check if space available on east side of inner courtyard. Place if possible. b. If not placed, check for space along east wall of pinwheel and place if possible. c. If not, check for space along east wing wall and place if possible. d. If not placed, check for available space on the east side of the north wing wall. Place if possible e. If not, display the default screen.
6	West OR West	West None	a. Check if space available on the west side of the inner courtyard and place if possible. b. If not placed, check west side of pinwheel for placement space. Place, if possible. c. If not, check for space available on west wing wall and place if possible. d. If not, check for space along the west side of the south wing hall. Place if possible. e. If not placed, display the default screen.

Scheme Number	Wall Assignment Placed	Wall Assignment Candidate	Guidelines
7	East	West	a. Check for available space on the west side of the south wing wall. Place if possible.
			b. If not placed. follow the procedures outlined in Scheme 6, steps a through c and place if possible.
			c. If not. display the default screen.
8	West	East	a. Check for available space on the east side of the north wing wall and place if possible.
			b. If not placed, follow the procedures described in Scheme 5, steps a through c and place if possible.
			c. If not placed, display the default screen.
9	West	North	a. Check if space available inside the north side courtyard area and place if possible.
			b. If not, check if space available on north side of pinwheel and place if possible.
			c. If not, check if space available along the north wing wall and place if possible.
			d. If not placed, check for space on the north side of west wing wall and place if possible.
			e. If not, display the default screen and allow the user to place activity.
10	East	North	a. Follow the procedures outlined in Scheme 9.
11	West	South	a. Check if space available inside the south side of the courtyard and place if possible.
			b. If not placed, check for space on the south side of the pinwheel and place if possible.
			c. If not, check for available space on the south wing wall and place if possible.

Scheme Number	Wall Assignment Placed	Wall Assignment Candidate	Guidelines
			d. If not placed, check for space along the south side of the northeast wing wall and place if possible.
			e. If not, display default screen.
12	East	South	a. Follow the procedures outlined in Scheme 11.
13	South	West	a. Check if space available on the west side of the inner courtyard and place if possible.
			b. If not placed, check west side of pinwheel for placement space. Place, if possible.
			c. If not, check for space available on west wing wall and place if possible.
			d. If not, check for space along the west side of the south wing hall. Place if possible.
			e. If not placed, display the default screen.
14	North	West	a. Follow the procedures outlined in Scheme 13.
15	North	East	a. Check if space available on east side of inner courtyard. Place if possible.
			b. If not placed, check for space along east wall of pinwheel and place if possible.
			c. If not, check for space along east wing wall and place if possible.
			d. If not placed, check for available space on the east side of the north wing wall. Place if possible.
			e. If not, display the default screen.
16	South	East	a. Follow the procedures outlined in Scheme 15.

VITA

Richard Edward Webb

Candidate for the Degree of

Doctor of Philosophy

Thesis: INTERACTIVE COMPUTER MODELING OF THE BUILDING DESIGN DEVELOPMENT PROCESS

Major Field: Industrial Engineering and Management

Biographical:

Personal Data: Born in Joplin, Missouri, March 5, 1955, the son of William Edward Webb and Joyce Wallace Webb. Married to Sandra Gayle Moats on December 27, 1976. Father of Megan Ann Webb, born on September 3, 1979 and Jaclyn Leigh Webb, born on April 20, 1982.

Education: Graduated from Nathan Hale High School, Tulsa, Oklahoma, in May, 1973; received Bachelor of Science and Architectural Studies from Oklahoma State University in May, 1977; received Master of Science Degree from Oklahoma State University in May, 1979, with a major in Industrial Engineering; completed requirements for the Doctor of Philosophy degree at Oklahoma State University in July, 1984.

Professional Experience: Management Consultant, Arthur Young and Company, June 1979 until January 1981; Management Consultant, Webb, Shirley & Box, January 1984, to present; Registered Professional Engineer in the state of Oklahoma.