AN EXPERIMENTAL STUDY OF THE MAN-MACHINE INTERFACE

by

PAUL STANTON MASULIS
B.S., Carnegie-Mellon University, 1976

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE IN BUSINESS ADMINISTRATION

in

THE FACULTY OF GRADUATE STUDIES
(Commerce and Business Administration)

We accept this thesis as conforming
to the required standard:

THE UNIVERSITY OF BRITISH COLUMBIA
May, 1978

Paul Stanton Masulis

Faculty of Commerce and Business Administration

The University of British Columbia
2075 Wesbrook Place
Vancouver, Canada
V6T 1J6

Date: May 25, 1978

ABSTRACT

In this thesis, the author pursued two objectives. The first objective was to present a working example of a convenient, "idiot-proof", interactive computer program (designed with the user - not the programmer - in mind). The second objective was to investigate how various types of users interact with the computer, with the intention of reaching some conclusions about which program interfaces were most appropriate and convenient for various user types. In addition, some theories about the effects of various behavioural variables were investigated.

The experimental tool used for this research was a simple interactive computer game in which the participants searched for the optimum profit in a three-dimensional space, given a fixed time limit. Frequent periodic measurements were automatically collected on user performance, attitude, requests for reports, utilization of special features, and other variables; also, the solution protocol of each participant was recorded. The users were categorized by cognitive style (heuristic/analytic), risk attitude, and previous computer experience as determined by a battery of pre-tests and questionnaires.

In analyzing the results, it was found that experience level was the dominating factor on all dimensions: novices were slower, finished less frequently, and were significantly less confident than experienced players. A highly structured program interface was found to be more appropriate for these

new users. Experience was also the dominating factor in the use of reports, although novices did show a marked learning effect over time - as did all users on most dimensions of performance and behaviour. As previously hypothesized, analytic-types and risk-takers played significantly faster and were more confident than heuristic-types and risk-averters, respectively.

Concerning utilization of special program features, it was found that input response defaults influenced users in unfamiliar situations (ones which were new or did not have clear-cut responses), and didn't affect them at all in familiar circumstances. Analytic-types made least use of defaults. Risk-averters were least likely to abbreviate commands. Also, the extent to which commands were abbreviated depended much upon their length. Finally, in the area of solution protocols, it was indeed found that heuristic-types were much less structured in their approach to solving the problem than analytic-types.

TABLE OF CONTENTS

LIST OF TABLES

ACKNOWLEDGEMENT

I would like to take this opportunity to thank my advisors, Professors Albert Dexter and Izak Benbasat, for their role in the establishment and completion of this research: for summer employment in 1977 as research assistant, doing systems programming for two interactive computer games (work which first instilled my desire to investigate more carefully the man-machine interface); for suggestions regarding implementation of this research; for occasional administrative support; and especially for their contributions to this final document. I also thank my third committee member, Prof. Ronald Taylor, for his additional assistance and enthusiasm.

In addition, I owe sincere thanks to many of the students of Saint Andrew's Residence Hall for their participation, support, encouragement, and fellowship during this research and throughout my two years at the University of British Columbia.

Chapter One

INTRODUCTION

Even with the increasing programming effort going into the development of interactive programs (especially simulations and games), design of the actual man-machine interface has continued to be neglected.

Many of the interactive programs which this author has encountered tended to be quite frustrating to use. For example, some required that entire commands be spelled out, when one or two letters would be unambiguous. Others required input in a fixed format. Still others responded to an illegal input with an unintelligible system error message. It was not necessary to look very far for examples; they were quite prolific in the system program libraries of every computer installation visited. Some examples from the literature are illustrated in chapter two.

A review of the literature indicated that insufficient attention seems to have been paid to this issue. Computer games abound as research tools, but few researchers appear to have considered whether their man-computer interface significantly biased or discouraged their subjects.

Perusal of the standard texts on man-computer communication was also quite frustrating; the topics covered were often too general or sophisticated for the designer of interactive programs for normal CRT terminals. Even in the most useful chapter, dozens of interface designs were listed and described, but few hints were given as to when each was appropriate. Also, few references were suggested for seeking further details (reinforcing the notion that this area had been forgotten in the literature).

The goal of this thesis was to examine a few interface designs experimentally, with the intention of determining the conditions under which each was most appropriate and indicating any forms which may bias the user's behaviour. A lesser goal was to also study the effects of some behavioural variables upon performance, attitude, and solution protocol.

In the following pages, the relevant literature is first briefly reviewed. Then, the actual program code is presented and analyzed, with particular attention paid to the input prompts, the methods for accepting input from the user, and the techniques for detecting and handling user input errors (often referred to as "idiot-proofing").

Next, the actual data collection and analysis are described; then the following chapter presents and discusses the results relating to user performance, use of program defaults and command abbreviations, behaviour over time, and the participants' solution protocols.

Finally, some practical implications and future directions for research are suggested in the concluding chapter.

Chapter Two

LITERATURE REVIEW

Background and Motivation

In this chapter, some of the literature relevant to this research is presented: background review, user engineering articles, previous computer experimentation involving behavioural variables, literature concerning problem solution protocols, and texts on the man-machine interface are all discussed. Before looking at the literature which has direct bearing upon this work, however, an indirectly related reference is mentioned.

Although man-machine communication has only recently received serious attention, it has had a very interesting history. In his book Systems Psychology,[1] Kenyon B. DeGreene provided a very good summary of its history, from the intensive development of computer equipment in the 1950s, to computer programs in the early 1960s and man-machine interrelationships in the late 1960s. Chapter 10, entitled "Man-Machine Interrelationships," was described very well in its own introduction:

> This chapter first reviews history and trends toward greater computer systematization. Areas of spatial and temporal interface between man and computer receive special attention. We then consider important specialized areas of research and application, which include means of direct, usually dynamic man-computer communication by input and display devices in terms of given language structures, time-sharing, and "symbiotic" problem solving. Human factors and managerial considerations in computer systems follow. The chapter ends with an evaluation of the continued societal impact of computers.[2]

It also included a useful section on the main sources of design and operational error in computer systems. Overall, this reference provides a good background for many of the ideas presented and practiced in this thesis.

More directly related to this research is a doctoral thesis by Peter G. W. Keen[3] at Harvard University; in fact, it is probably the single major cause of this research. In his thesis, Keen suggested an interactive computer simulation program which allowed the user nearly complete freedom to decide what he would like to do next, instead of the traditional 'request input-simulate-display output-repeat' cycle. Hypothesizing that this concept had rather strong implications for ease of use by inexperienced computer users and possibly by those who display a non-analytic cognitive style of problem solving, this author decided to experimentally test the implications of Keen's suggestions. It should be mentioned, however, that this author also considers the concept of less structured computer interfaces very important - in the proper environment. For instance, they would be appropriate for programs which are run frequently, by experienced users. (Since this research was begun, it has been learned that Botkin[4] found that such an unstructured model was used with equal effectiveness by both analytic and heuristic decision makers).

The Inventory Management Game, a research tool used quite extensively by Benbasat[5] and others at the University of British Columbia, was another influential cause of this thesis. Experience with the original version of the computer game demonstrated that it unnecessarily neglected the user. For instance, all responses - including 'YES' and 'NO' - had to be typed in full; no reasonable default values were provided to minimize routine typing; some input requests were ambiguous; and any typing errors were answered by the unintelligible system message "ILLEGAL CHARACTER, ENTER REPLACEMENT NUMBER, OR RE-ENTER REST OF LINE FROM POINT OF ERROR, OR 'MTS.'" A new version of the Inventory Management Game has corrected many of these shortcomings, and the authors have provided some interesting results relating some characteristics of an information system and a decison maker to the resulting decision making performance. For details, consult the papers by Benbasat and Schroeder[6], Benbasat and Taylor[7], and Benbasat and Dexter[8].

Another early example of a non-user-oriented interactive computer program appeared in the June, 1969 issue of Management Science.[9] The authors stated in the introduction that one motivation for their research was to answer the question "How should a problem environment be structured in order to effectively employ the abilities of both the manager and the on-line, realtime computer?" They then proceeded to describe a job-shop simulation program which required the user to type such non-mnemonic commands as "FORSIM=2*" to continue simulating or "SRULE=3, HRULE=6, HRS=80, QZ=-201*" to change parameters, and which outputs a table with the ambiguous headings "L, M, J, I, NEXT, KACT, PROM, LEFT, CUSH, LIPR, COMP, SETUP, IQ." To this author, this just was not a convincing effort to "effectively employ" the abilities of both manager and computer.

User Engineering Methods

One of the outcomes of the 1973 National Computer Conference was an excellent article by Anthony Wasserman,[10] entitled "The Design of 'Idiot-Proof' Interactive Programs." According to Wasserman, a program is said to be idiot-proof if

> it is designed to anticipate any possible action by its users and to respond in such a manner as to minimize the chances of program or system failure while shielding the user from the effects of such a failure.[11]

Bearing in mind Murphy's Law - anything that can possibly go wrong will go wrong - Wasserman suggested five principles:

1. Provide a program action for every possible type of input.
2. Minimize the need for the user to learn about the Computer System.
3. Provide a large number of explicit diagnostics, along with extensive on-line user assistance.
4. Provide program short-cuts for knowledgeable users.
5. Allow the user to express the same message in more than one way.

These principles were all described in detail in the article, which then concluded with the following statement:

> There is a serious need for improved facilities for the design of idiot-proof interactive programs. With a growing number of non-programmers using computers, development of comfortable man-machine interfaces will outweigh many traditional considerations in the overall creation of interactive programs.[12]

As stated earlier, the purpose of this thesis was, in fact, to provide a working example of an idiot-proof program and to perhaps make some contribution to the above-mentioned need.

Another excellent article about idiot-proofing (or user engineering, or error engineering) came out of the 1971 Fall Joint Computer Conference. In it, Wilfred J. Hansen[13] suggested four user engineering principles:

- Know the user
- Minimize memorization
  - Selection not entry
  - Names not numbers
  - Predictable behavior
  - Access to system information
- Optimize operations
  - Rapid execution of common operations
  - Display inertia
  - Muscle memory
  - Reorganize command parameters
- Engineer for errors
  - Good error messages
  - Engineer out the common errors
  - Reversible actions
  - Redundancy
  - Data structure integrity[14]

Since some of these principles are quite terse, descriptions of a few of the more vague ones follow: 'Names not numbers' suggests that users be allowed to enter actual names rather than associated number codes; 'Predictable behavior' suggests that the program have a "personality" and be consistent in its output display and input requirements; 'Display inertia' suggests that the terminal display should change as little as necessary in carrying out requests; and 'Muscle memory' suggests a need to design a system so that repetitive operations can be delegated to the lower part of the brain (in the same way as many of the operations in driving and typing). The article also provided an excellent example of a user-engineered program.

Previous Experimentation with Behavioural Variables

Another reason for this research was to relate some behavioural aspects of users to their reactions to various program features. This was not a new concept: For instance, K. D. Eason[15] performed a study of "The Manager as a Computer User." In it, the nature of management was presented, then a survey of 200 computer users was described, and, finally, four major causes for user dissatisfaction with computer systems were analyzed. The four causes were: an inadequate match to the manager's needs, new problems caused by system advancement, changes in user expectations (as they realize the computer's potential), and lack of both time and desire to learn how to operate complex systems. Eason found that computer programs would have to be more convenient and more flexible in the future; he concluded that "unless it is possible to design forms of interaction acceptable to managers, this role for the manager may be very short lived."[16]

A similar study is described in an article entitled "Human Factors Evaluation of a Computer Based Information Storage and Retrieval System."[17] The authors evaluated a government computer system called the Central Information Reference and Control (CIRC) system, and found that:

> In reviewing the results from the evaluation, there appeared to be three main factors which influence an individual's satisfaction with the CIRC system: (1) training and level of proficiency, (2) amount of information in the system to meet task requirements, and (3) the individual's tolerance for irrelevant material.[18]

The third point is particularly interesting. It is worthwhile to mention that one of the advantages of interactive systems is the potential to let the user choose what he needs - no more, no less; this may be a justification for providing more unstructured program interfaces which always allow the user to decide what he needs next.

In an article from Data Base, Theodore J. Mock[19] described "A Longitudinal Study of Some Information Structure Alternatives." Mock studied user performance with various Accounting Information System models, with the objective of considering the impact of several behavioural variables and technical information structure variables upon decision makers' profit performance and learning patterns.

> In summary, the first set of experiments did demonstrate the feasibility of experimentally investigating expected differences in information structures and the impact of certain behavioural variables... Experimental data which implies the significance of behavioural factors increases validity of suggesting tailorized information systems for decison makers exhibiting different behavioural characteristics.[20]

Another study, by Wynne and Dickson,[21] looked at "Experienced Managers' Performance in Experimental Man-Machine Decision System Simulation." They were concerned with the effectiveness of Man-Machine Decision Information Systems (MMDIS), and ran experiments using an interactive simulation program (which, unfortunately, was not explicitly described in the article). As a result of their research, Wynne and Dicksen reached two main conclusions:

> First, the differential performance of subjects is related not only to personality variables but also to information acquisition and usage patterns... It appears from work thus far that personality and cognitive style impact the effectiveness of MMDIS through the strategy of system usage by the human. Second, the effectiveness of an MMDIS must then be a function of the ease (or difficulty) with which the interactive computer program enables a decison maker to implement his preferred information handling strategy.[22]

In a paper entitled "The Impact of Cognitive Styles on Information System Design,"[23] Benbasat and Taylor suggested the following three generalizations:

> 1. Analytic decison-maker types tend to prefer decision aids and reporting systems which are quantitative in nature with results supported with mathematical formulas.
>
> 2. Heuristic decision-makers need to have more data search capabilities prior to reaching decisions. Since they rely on feedback and trial and error, an information system capability which can highlight trends and provide period by period comparisons would be suitable for them. The information system should give them capabilities to try alternative solutions and analyze the possible outcomes before they decide on their final approach to solving the problem.
>
> 3. Decision-makers are also different in terms of their data gathering styles. The preceptives would want a system which has capabilities of organizing and aggregating data into categories according to given parameters and exception reporting aids, whereas the receptives or maximal data users prefer an information system which has access to every piece of historical data.[24]

Turning now to toward the area of risk attitude, a paper by Taylor and Dunnette[25] contained an interesting result:

> Although risk-taking propensity influenced heavily both the amount of information processed and decision latency, it does not appear that high risk-takers attain faster decisions by processing each item of information more rapidly... Rather, it would appear that they are quite deliberate in

> attempting to extract as much value as possible from the smaller set of information they examine.[26]

These results were from an individually and manually administered decision simulation; the research of this thesis provided an opportunity to consider the same hypotheses in a computer environment. In another paper, on psychological determinants of bounded rationality, Taylor[27] provides examples of more risk attitude studies.

<u>Problem Solution Protocols</u>

In the area of solution protocols (also briefly considered in this thesis), Barrett's description of cognitive style decision approaches[28] was directly relevant. Barrett compared heuristic and analytic decision styles on five dimensions. For example, with regard to learning, it was said that heuristics learned more by acting and placed emphasis on feedback; analytics learned more by analyzing and placed less emphasis on feedback. In the area of search strategy, heuristics used trial and error, while analytics used formal rational analysis. Finally, regarding approach to analysis, heuristics used common sense, intuition, and feelings, whereas analytics developed explicit models of the situation.

<u>The Man-Machine Interface</u>

It is reiterated at this time, that although this thesis looked at many of the concepts mentioned throughout this chapter, its original purpose was to study the man-machine interface at a fairly low level, with the objective of reaching some conclusions about which interactive programming techniques are most helpful for users, and least likely to bias their behaviour. This appeared to be an original area of research and, as stated earlier, was quite untouched in the literature. Many articles and books existed which suggested the philosophies of various individuals and described working prototype systems, but few have experimentally tested the implications of their techniques for user performance and behaviour.

The standard texts, Man-Machine Communication by Meadow[29] and Design of Man-Computer Dialogues by Martin,[30] provided some assistance, although they were often too general or sophisticated to be of direct assistance in normal, day-to-day situations. Martin's text, found by this author to be the more practical of the two, did have one particularly relevant chapter: chapter seven explicitly considered display methods for alphanumeric computer terminals with TV-like screens:

> In tackling an application, the systems analyst must make some basic decisions about the structure of the screen conversation...Twenty-three techniques of conversation are illustrated below. They have been given the names:
> 1. Simple query
> 2. Mnemonic techniques
> 3. English-language techniques
> 4. Programming-like statements
> 5. Action code systems
> 6. Multiple action code systems
> 7. Building up a record
> 8. Scroll techniques
> 9. Simple instruction to operator
> 10. Multiple instruction to operator
> 11. Menu selection
> 12. Multiscreen menu
> 13. Telephone directory technique
> 14. Multipart menu

15. Multianswer menu
16. Use of displayed formats
17. Variable-length multiple entry
18. Multiple-format statements
19. Form filling
20. Overwriting
21. Panel modification techniques
22. Text-editting techniques
23. Hybrid dialogue[31]

Martin then proceeded to describe each of these methods in very good detail but, unfortunately, too seldom really indicated when each was appropriate. So the systems analyst finds himself barraged with twenty-three very simple to very complex methods of designing a terminal interface, and can only guess which is most appropriate for his situation.

Throughout the remainder of this thesis, a personal philosophy of man-machine interface design is presented, and the effects of a small set of man-machine interface techniques upon various user types are investigated.

Chapter Three

THE COMPUTER PROGRAM

Program Description

The primary tool for this research was a simple interactive computer game. Details of the game will be provided in the next chapter; the user engineering aspects of the computer program are described next.

The computer game was completely written in FORTRAN (a listing of the code appears in Appendix A). The actual game is only a small part of the program; a significant amount of programming was necessary to achieve the desired user interface and collect all the required data. It was also necessary to use a few subroutines from the University of British Columbia Computing Centre subprogram library, including timing routines, file control routines, and a character comparison routine.

The game had two distinct versions, both described in detail in chapter four. Briefly, one version was highly structured and led the user through the simulation by looping through a set of questions; and the other version was rather unstructured and expected the user to lead the simulation by entering commands in any order he liked. To facilitate this dual version concept, the program had to be highly modular. The program was made up of a brief main program, which called one of two "control" subroutines (to get the appropriate version), which in turn called a number of the remaining ten subroutines.

One of the ten subroutines, READPF, was called at the program startup to read in the profit function (a 30 by 70 matrix). Another subroutine, SIMUL, performed the actual simulation of another period (i.e. another trial). Three of the subroutines, GETLIN, GETLIT, and GETNUM, handled all terminal input. One subroutine, OUTMES, was just a collection of all output messages needed throughout the program; by gathering them in one place, only one routine needed to be recompiled whenever the user interface was refined. Three more routines, HISTRY, SORTH, and SGRAPH, displayed the three available reports. Finally, the remaining routine, ZEND, performed all end-of-game cleanup.

In the structured version of the game, the control subroutine simply called the appropriate subroutines in a prescribed order, as a continuous loop. In the unstructured version, the program waited for a command from the user, decoded it, and called the specified routine. Hence, the only extra programming effort required in order to provide two versions lay in the two (quite straightforward) control routines.

User Engineering Methods

The remainder of this chapter describes the user engineering aspects of the program. Although the approach is a personal one developed through years of experience, the reader will note that the methods satisfy many of the criteria and suggestions of Wasserman and Hansen presented in chapter two.

In designing the actual input of responses and commands from the user, ease of use was given top priority. First, the need for memorization by users was minimized. In the unstructured version, a list of all commands - and brief descriptions of them - was available anytime. In both versions, all input prompts were of the same format, illustrated by the following example:

Enter desired price level (1-30) [10] :

As can be seen, first the question was asked, then the allowed response range was indicated in parentheses, then the default value was indicated in brackets. The default value was the value which the computer would assume the user wanted if he entered nothing else (in this case, the price which he had chosen in the previous period), and was included in order to reduce routine typing. Finally, to further eliminate the need for memorization, complete histories of all previous activity were available to the user at anytime.

The program also handled all input processing itself. This way, all user errors were intercepted by the program ("idiot-proofing") before the system software could find it and respond with some illegible message or interrupt. In this game, errors were responded to by the simple statement

Incorrect Input. Please Re-enter.

This was followed by a repeat of the original prompt, which of course reminded the user of the question, the allowed responses, and the current default value.

The program read all input as a string of alphanumeric characters (up to 60 of them). The string was scanned, character by character, up to the first blank or comma, and that substring was considered to be the response. If there were more characters following the blank (or comma), they would be used as the response to the next prompt(s) - allowing experienced users to type ahead, and save time and frustration. If the original prompt wanted an alphabetic response, then only the first character was used (since all commands and responses in this game begin with with different letters) - thereby permitting unlimited abbreviation. If the original prompt wanted a numeric response, then the program converted the substring to a number; admittedly, this was awkward in FORTRAN, but still inexpensive and well worthwhile for the user.

Finally, the user needed to be protected not only from himself, but from the computer and the environment in general. In the event of a computer crash or other major problem, the program had a save/restart facility. As the program ran, it wrote out a simple file. If anything caused the program to halt, a special run parameter allowed the user to restart the program at exactly where he left off - as if nothing had happened.

It is quite apparent that these features did not come without a cost. However, there is no reason why the input processing routines could not have been designed as a package to be linked with all other application programs needing interactive capabilities - an approach which would likely save programming costs in the future. Also, the routines could be written in a more appropriate language (probably assembler), to increase their efficiency. This is not to say that efficiency is a critical issue. Indeed, in most cases, the amount of time a program spends processing user input would likely be only a small part of the total cost of running any program, while the savings in user's time and frustration could be quite substantial.

Chapter Four

DATA COLLECTION METHODOLOGY

Pre-testing and Classification

The actual data collection for this research involved obtaining participants, pre-testing them, arranging for them to play the computer game, and automatically (by computer) collecting data on them as they played.

There were fifty participants in the experiment, virtually all of whom were students, and all of them volunteers (some lured by the possibility of winning one of five cash prizes). As was desired, the participants were quite diverse: some were undergraduates, others were graduates; some had extensive experience with computers, while others had never been near one; some were from commerce programmes, others from engineering, and still others from arts.

The experience difference was a crucial one to this experiment (necessary for testing the main hypotheses). As part of the pre-testing for the game, participants completed a short questionnaire about their history of contact with computers. As a result of this questionnaire, which simply asked people their year, faculty, number of times they had used computers via punched cards, and number of times they had used on-line computer terminals, they were classified as experienced or inexperienced (novice) users of on-line computer systems. However, bear in mind throughout this thesis that the experience effect may be somewhat confounded:

experienced computer users often also had more advanced mathematical training than novices.

Next, the Group Embedded Figures Test,[32] a timed pencil and paper test, was administered to the participants. The score on this test provided a indication of whether the participants displayed heuristic or analytic cognitive styles (see chapter two for definitions of these terms). For purposes of this research, the group was divided at its mean (15 on a scale of 18). Since this is a rather high division value, it is more appropriate to say that this research compares low and high analytics rather than pure heuristics and analytics.

Finally, the participants completed the Kogan and Wallach risk questionnaire.[33] Their score on the questionnaire provided a measure of their risk attitude; again, the group was split at its mean (30 on a scale of 60) and classified as risk-takers or risk-averters.

All of the above pre-testing was administered to groups of about ten over a three day period, and each session took just over 30 minutes to complete. As each subject left the pre-testing session, he selected a convenient time to play the computer game during the following week. Also as they left the pre-testing session, participants were given a brief set of instructions (see Appendix B). These instructions did not explain the nature of the computer game; rather, they provided directions for using the computer terminals and special program features. There was a separate set of instructions for each of the two versions of the game to which the subjects had been randomly assigned.

Administering the Game

The actual process of playing and administering the game is described in this section. Throughout this discussion, the sample interactions which appear in Appendix C may be consulted for clarification of any vague points.

The participants played the computer game in groups of three over the course of one week. The game had a maximum time limit of 30 minutes, after which it terminated automatically. About one-half of the subjects finished before exceeding the time limit.

When playing the game, each participant was instructed at the beginning that he was the manager of a one-product company and that he was expected to seek the optimum quantity of product to manufacture and the price to sell it for; that is, he was to maximize his company's profit. Hence, playing the game involved repeatedly setting different <price, quantity> combinations and simulating the next period to get the resulting profit.

If a subject found the maximum profit within 30 minutes, the game "rewarded" him by informing him with all manner of bells and whistles: the terminal screen filled up with dollar signs and congratulated the player, while the terminal bell beeped until stopped by the game administrator. Actually, the same bells and whistles announced an apology to those who ran out of time. This not only served to attract the game administrator's attention, but appeared to both amaze and please the participants.

As the game proceeded, the participants had access to any or all of three reports. The first was a simple history of their decisions and profits for the previous 25 periods. The second was also a history report, except that it was ordered by decreasing profit. Finally, the third was a 3-dimensional graph which displayed Profit/10 (i.e. one digit) for each <price, quantity> pair simulated thus far.

There are two sample game interactions in Appendix C, one for each of the two game versions. In the structured version, the user was essentially taken by the hand, and led through the game, step by step, in a predefined order. In the unstructured version, the user had more freedom to proceed as he wished by entering any of six commands (to set price or quantity, simulate another period, or look at the reports). It should be mentioned that none of the subjects had any difficulty in using the structured game version. In addition, participants with previous computer experience had no problems with the unstructured version. However, novice subjects often needed verbal assistance from the game administrator in order to get started with the unstructured version.

The profit function which the users were attempting to maximize appears in Appendix D. The basic profit function was the same for everyone; however, the <price, quantity> position of the optimum profit was generated randomly at program startup. Thus, for each participant, the optimum profit occured at a randomly set price between 5 and 25, and at a quantity between 15 and 55. Since the function did not change shape, but only moved, and since each person could search anywhere he wished, these steps should not have made the game more difficult for some subjects. Also, the profit values were scaled by another randomly generated constant to values between 70 and 99. These steps essentially made the game different for each participant and therefore eliminated any possibility of collusion.

Data Collection

As the participants played the game, the program automatically collected data about their performance and use of program features. For each period, information was recorded about: the amount of time taken to complete it; the chosen price, quantity, and resulting profit; number of commands executed; number of defaults taken; number of errors made; extent of input abbreviation; amount of use made of typeahead option; utilization of each report; and other aspects. A listing of a sample output file for one participant appears in Appendix E.

Additional data was collected about each participant's attitude as he played. After periods 5, 10, 15, ... (note line code 2 in the sample data file in Appendix E), the normal flow of activity in the game was interrupted by a brief questionnaire to get the user's confidence level, rating of the program usability, and enjoyment level (see the sample interactions in Appendix C).

The data collected also contained a machine-readable solution protocol for each participant, indicating exactly how each subject moved through the two-dimensional space in search of the optimum profit. It was found that by plotting the <price, quantity> pairs in order of simulation (as in Appendix F) and then connecting the dots, one could detect whether users employed a random search, a structured trial and error, or a binary search or other well-defined algorithm, all of which will be discussed more completely in chapter six.

In the remainder of this thesis, the output results for the fifty participants are presented and discussed.

Chapter Five

THE HYPOTHESES

Introduction

Before the final results are analyzed, this chapter briefly introduces the hypotheses which were being tested. Although the data from this game provides numerous possibilities for analysis, the 26 hypotheses of this chapter were the major motivations for this research and will receive most of the attention throughout the remainder of this thesis. Since this is exploratory research, some of the hypotheses have no strong theoretical basis; however, other hypotheses do attempt to verify the findings of others. In this chapter, the hypotheses will simply be stated, with detailed analysis and connection to previous research to follow in the next chapter.

In nearly all of the hypotheses, there are four independent variables, each at two levels: game version (structured or unstructured), experience level (novice or experienced), cognitive style (low analytic or high analytic), and risk attitude (risk-averter or risk-taker). For simplicity, these variables will be called Mode, Exp, Style, and Risk, respectively.

Performance and Game Version

The first category of hypotheses is related to general user performance and the two game versions. The first hypothesis is rather special, and is assigned the number zero to differentiate it from the rest.

Hypothesis 0 - Everyone will enjoy playing the game.

Hypothesis 1 - Mode, Exp, Style, and Risk will all affect the average time spent playing each period.

Hypothesis 2 - Mode, Exp, Style, and Risk will all affect whether the subjects finish within the 30 minute time limit.

Hypothesis 3 - Mode, Exp, Style, and Risk will all affect the average confidence level of the participants.

Hypothesis 4 - Unstructured game version players will be faster, finish more often, and be more confident than structured version players.

Hypothesis 5 - Experienced players will be faster, finish more often, and be more confident than novices.

Hypothesis 6 - High analytics will be faster, finish more often, and be more confident than low analytics.

Hypothesis 7 - Risk-takers will be faster, finish more often, and be more confident than risk-averters.

Hypothesis 8 - The Mode/Exp interaction will affect the error rate of the participants.

Special Program Features

The next category of hypotheses is related to the use of special program features, such as default values and abbreviations.

Hypothesis 9 - The default values for price and quantity (at the beginning of the game) will influence most users.

Hypothesis 10 - Setting the default response for questions (about the user's desire to see various reports) to 'yes' rather than 'no' will not influence the participant's actual response.

Hypothesis 11 - Exp, Style, and Risk will all affect whether users accept default values (when appropriate).

Hypothesis 12 - Mode, Exp, Style, and Risk will all affect the extent to which users abbreviate commands.

Hypothesis 13 - The length of commands will be the main factor affecting the extent to which they are abbreviated by users.

Comparisons over Time

The third category of hypotheses is with regard to comparisons over time, and indicates expected differences between behaviour during the beginning of the game and during the remainder of the game.

Hypothesis 14 - Average time spent playing each period will decrease with time.

Hypothesis 15 - User confidence will increase with time.

Hypothesis 16 - User ratings of the usability of the computer program will improve with time.

Hypothesis 17 - The extent of abbreviation by unstructured game players will increase with time.

Hypothesis 18 - Usage of History reports will decrease with time.

Hypothesis 19 - Usage of Ordered History reports will decrease with time.

Hypothesis 20 - Usage of Graphs will increase with time.

Report Usage and Solution Protocols

These last five hypotheses concern either the usage of reports or solution protocol dimensions.

Hypothesis 21 - Mode, Exp, and Style will all affect the use of History reports.

Hypothesis 22 - Mode, Exp, and Style will all affect the use of Ordered History reports.

Hypothesis 23 - Mode, Exp, and Style will all affect the use of Graphs.

Hypothesis 24 - Exp, Style, and Risk will all affect whether users displayed a structured approach to solving the problem (with the emphasis on Style).

Hypothesis 25 - Exp, Style, and Risk will all affect the amount of dispersion displayed in the search for the optimum (with the emphasis again on Style).

The last two hypotheses are explained in more detail in chapter six.

The results of the tests of all these hypotheses are presented and analyzed in the next chapter. A summary of the results appears in Appendix G.

Chapter Six

ANALYSIS OF RESULTS

Data Preparation

Before the game data could be statistically analyzed, it had to be converted to a more convenient form. Thus the output files for each of the 50 participants were compressed into one line each, yielding one file with 50 very long lines.

To derive this new file, some variables were simply copied directly from the original file, others were summations of original data (for example, total time playing the game), others were averages (user confidence), others were extractions (minutes per period for the first 10 periods), and still others were results normalized to 100 (for example, the number of graphs requested per 100 periods).

As mentioned in chapter five, nearly all of the hypotheses involve the following four two-level variables: game version (1=structured, 2=unstructured), experience level (1=novice, 2=experienced), cognitive style (1=low analytic, 2=high analytic), and risk attitude (1=risk-averter, 2=risk-taker). Again, for simplicity, these variables will be referred to as Mode, Exp, Style, and Risk, respectively.

Statistical Analysis

Three basic types of analysis were performed in this analysis, all of them using the Statistical Package for the Social Sciences (SPSS).[34] Since most hypotheses were concerned with determining which factors (independent variables) most affected a given game outcome (dependent variable), an analysis of variance (ANOVA) was employed to test these hypotheses, using the ANOVA routine in SPSS. In most cases, either a three-way or four-way classification (with three-way and four-way interactions assumed to be zero) was used. The general model for the three-way classification was

$$y = a + b1x1 + b2x2 + b3x3 + c1x1x2 + c2x1x3 + c3x2x3 + e$$

where y was the dependent variable, a was the overall mean, xN were the independent variables, bN were the main effects, cN were the interaction effects, and e was the error term. The model for the four-way classification was the same, except with four main effects and six interaction effects.

Other hypotheses were concerned with how two groups of subjects differed on an individual variable. In these cases, two mean values were to be compared, so one-tailed t-tests were used to test the hypothesized relationships. The SPSS T-TEST routine, with cases classified into two groups, was used to perform the test, using a pooled variance (since the two population variances were assumed to be different). The remaining hypotheses (all related to Comparisons over Time) involved the comparison of two variables over all subjects; paired t-tests were employed to test these hypotheses. Again, the SPSS T-TEST routine was used to perform the test; however, this time paired observations were specified.

In the analyses to follow, the SPSS results are reproduced in their standard formats. The analysis of variance tables display the main effects and the 2-way interactions (expressed as "variable/variable").

Hypotheses about Performance

The results concerning hypothesis 0 - Everyone will enjoy playing the game - were especially encouraging. Throughout the entire game, the mean enjoyment level for all players was 7.0 on a scale of 1 to 9, where 1=bored and 9=enjoying the game (see the sample attitude questionnaire in appendix C). This was important because it added credence to the game results: subjects did not just go through the motions to get the game over with; they actually enjoyed the game and quite probably "played to win." An analysis of variance was carried out to see whether any particular user types enjoyed the game more than others. As can be seen in table 1, none of the four independent variables was significant; indeed, the overall significance level was only 0.71.

SOURCE VAR.	S.SQ.	DF.	M.SQ.	F	SIGNIF.
Mode	0.64	1	0.64	0.33	0.57
Exp	0.04	1	0.04	0.02	0.89
Style	0.35	1	0.35	0.18	0.67
Risk	0.47	1	0.47	0.24	0.63
Mode/Exp	0.40	1	0.40	0.20	0.65
Mode/Style	0.45	1	0.45	0.23	0.64
Mode/Risk	0.06	1	0.06	0.03	0.87
Exp/Style	2.08	1	2.08	1.06	0.31
Exp/Risk	9.08	1	9.08	4.64	0.04**
Style/Risk	0.24	1	0.24	0.12	0.73
Explained	13.82	10	1.38	0.71	0.71
Residual	76.18	39	1.95		
Total	90.00	49	1.84		

Table 1. ANOVA - Game Enjoyment

The analysis of hypothesis 1 - Mode, Exp, Style, and Risk will all affect the average time spent playing each period - is presented in table 2. Neither game version nor cognitive style impacted playing speed. However, risk attitude and experience both had a significant effect upon the number of minutes spent playing each period (these two factors will be investigated in more detail in hypotheses 5 and 7).

SOURCE VAR.	S.SQ.	DF.	M.SQ.	F	SIGNIF.
Mode	112.68	1	112.68	0.14	0.71
Exp	2321.79	1	2321.79	2.96	0.09**
Style	255.99	1	255.99	0.33	0.57
Risk	3154.83	1	3154.83	4.02	0.05**
Mode/Exp	168.27	1	168.27	0.21	0.65
Mode/Style	572.40	1	572.40	0.73	0.40
Mode/Risk	137.12	1	137.12	0.18	0.68
Exp/Style	225.70	1	225.70	0.29	0.60
Exp/Risk	2152.16	1	2152.16	2.74	0.11*
Style/Risk	348.61	1	348.61	0.44	0.51
Explained	10124.80	10	1012.48	1.29	0.27
Residual	30601.30	39	784.65		
Total	40726.10	49	831.14		

Table 2. ANOVA - Minutes/Period

Table 3 displays the analysis of variance for hypothesis 2 - Mode, Exp, Style, and Risk will all affect whether the subjects finish within the 30 minute time limit. Again, the results indicated that game version had no effect whatsoever. Cognitive style and risk attitude were also insignificant, while experience level was highly significant, indicating that game termination was almost completely determined by the experience level of the players (see hypothesis 5 for more details). This suggests that researchers should be extremely wary of this factor when carrying out experiments using on-line computer terminals.

SOURCE VAR.	S.SQ.	DF.	M.SQ.	F	SIGNIF.
Mode	0.00	1	0.00	0.00	1.00
Exp	3.28	1	3.28	16.75	0.00**
Style	0.29	1	0.29	1.46	0.23
Risk	0.21	1	0.21	1.08	0.31
Mode/Exp	0.05	1	0.05	0.24	0.62
Mode/Style	0.08	1	0.08	0.39	0.54
Mode/Risk	0.06	1	0.06	0.32	0.57
Exp/Style	0.33	1	0.33	1.73	0.20
Exp/Risk	0.14	1	0.14	0.73	0.40
Style/Risk	0.01	1	0.01	0.05	0.83
Explained	4.83	10	0.48	2.46	0.02
Residual	7.65	39	0.20		
Total	12.48	49	0.26		

Table 3. ANOVA - Termination on Time

Hypothesis 3 - Mode, Exp, Style, and Risk will all affect the average confidence level of the participants - was tested next. As the analysis (table 4) demonstrates, game version was once again highly insignificant; risk attitude had a weak level of significance. Experience again seemed to have a strong influence upon confidence, and cognitive style also appeared as an important factor (hypothesis 6 will investigate this further).

Having tested the three general hypotheses about performance, the next four hypotheses investigate this area at a more detailed level. The analysis for hypotheses 4 through 7 appears in table 5. While the previous ANOVAs indicated the relative importance of the factors when considered together,

SOURCE VAR.	S.SQ.	DF.	M.SQ.	F	SIGNIF.
Mode	1.60	1	1.60	0.01	0.95
Exp	2409.76	1	2409.76	6.99	0.01**
Style	1145.42	1	1145.42	3.32	0.08**
Risk	910.34	1	910.34	2.64	0.11*
Mode/Exp	23.79	1	23.79	0.07	0.79
Mode/Style	101.21	1	101.21	0.29	0.59
Mode/Risk	556.72	1	556.72	1.62	0.21
Exp/Style	172.51	1	172.51	0.50	0.48
Exp/Risk	9.51	1	9.51	0.03	0.87
Style/Risk	448.38	1	448.38	1.30	0.26
Explained	6678.06	10	667.81	1.94	0.07
Residual	13437.94	39	344.56		
Total	20116.00	49	410.53		

Table 4. ANOVA - Confidence Level

the t-tests to follow will test the hypothesized differences between groups on a single variable, and the directions of those differences. In table 5, the variable Term. (termination) is a two-level variable indicating whether people finished on time (Term.=0) or not (Term.=1). The playing speed variable, Min/Per., is the number of minutes spent playing each period. Finally, Confid. (user confidence) indicates the number of people (out of 100) whcm users thought were performing better than them (see the attitude questionnaire example in appendix C).

Hypothesis 4 - Unstructured game version players will be faster, finish more often, and be more confident than structured version players - was tested by the first 3 t-tests in table 5, where it was seen that this hypothesis was completely rejected. Consistent with the findings of the

VARIABLE	GROUPING	#	MEAN	STDEV.	T	PROB
Term.	Struct.	24	0.50	0.51		
	Unstruct.	26	0.46	0.51	0.27	0.34
Min/Per.	Struct.	24	0.77	0.33		
	Unstruct.	26	0.78	0.25	-0.22	0.41
Confid.	Struct.	24	40.88	24.35		
	Unstruct.	26	39.73	16.02	0.20	0.42
Term.	Novice	30	0.70	0.47		
	Exper.	20	0.15	0.37	4.44	0.00**
Min/Per.	Novice	30	0.84	0.33		
	Exper.	20	0.69	0.19	1.84	0.04**
Confid.	Novice	30	46.57	17.48		
	Exper.	20	30.85	20.89	2.88	0.00**
Term.	High-anal.	21	0.38	0.49		
	Low-anal.	29	0.62	0.50	-1.69	0.05**
Min/per.	High-anal.	21	0.74	0.24		
	Low-anal.	29	0.83	0.35	-1.06	0.15*
Confid.	High-anal.	21	34.83	20.67		
	Low-anal.	29	47.81	17.46	-2.34	0.01**
Term.	Risk-taker	19	0.37	0.50		
	R-averter	31	0.55	0.51	-1.23	0.11*
Min/Per.	Risk-taker	19	0.67	0.18		
	R-averter	31	0.84	0.32	-2.20	0.02**
Confid.	Risk-taker	19	33.42	16.65		
	R-averter	31	44.48	21.36	-1.93	0.03**

Table 5. T-TESTS - Performance and Structure

ANOVAs, game version had no significant impact upon the variables speed, termination, and confidence. This would seem to contradict the claim of the unstructured game version's superiority; however, it is quite likely that this game was just too simple to provide a significant difference in freedom between the two versions. Actually, observation by the game administrator, problems with starting novice participants playing, and verbal comments from the participants all indicated a greater difference than implied in table 5; novices appeared to have more trouble with the unstructured game than experienced players.

The fourth through sixth rows in table 5 tested hypothesis 5 - Experienced players will be faster, finish more often, and be more confident than novices. As expected, this hypothesis was strongly supported, again indicating that experience is a factor which should be seriously accounted for in all computer experiments. These results are similar to the findings of MacCrimmon,[35] who concluded that "experienced individuals seemed to be the most desirable subjects to utilize in decision making experiments and research."[36]

The analysis for hypothesis 6 - High analytics will be faster, finish more often, and be more confident than low analytics - was provided by t-tests 7 through 9. The hypothesis was only weakly significant on the speed variable, but termination and confidence both displayed highly significant differences between groups. These results were generally consistent with the conclusions of Benbasat and Taylor (see chapter two).

The test of hypothesis 7 - Risk-takers will be faster, finish more often, and be more confident than risk-averters - was provided by the last three tests in table 5. Speed and confidence showed very significant differences between groups, while termination was less significant. This would seem to contradict the findings of Taylor and Dunnette, especially with respect to time per period (see chapter two). This, however, needs further investigation since their research involved decision-making in a non-computerized environment.

The last hypothesis of this section tested the belief that novices would have difficulty with the unstructured game version, and would display it through an increased error rate. Hypothesis 8 - The Mode/Exp interaction will affect the error rate of the participants - was analyzed by the ANOVA in table 6. Clearly, there were no highly significant variables, and the hypothesis was rejected (it may well be that novices compensated any dificulties by devoting increased thought and care to each move they made, a possibility which was supported by the playing speed findings). The lack of significant effects was possibly caused by the fact that very few errors were made in the game. Out of 50 participants, only 11 made any errors anywhere in the game; seven low analytics averaged less than 5 errors per 100 periods, and four high analytics averaged 2 errors per 100 periods (where up to 6 inputs were entered each period).

Hypotheses about the use of Special Program Features

Like hypothesis 0, hypothesis 9 - The default values for price and quantity (at the beginning of the game) will influence most users - was tested by simple count. It was found in this experiment that 28 of the 50 participants accepted at least one of the opening default values (values

SOURCE VAR.	S.SQ.	DF.	M.SQ.	F	SIGNIF.
Mode	15.34	1	15.34	0.63	0.43
Exp	20.77	1	20.77	0.85	0.36
Style	66.26	1	66.26	2.70	0.11*
Risk	5.76	1	5.76	0.24	0.63
Mode/Exp	4.96	1	4.96	0.20	0.66
Mode/Style	22.15	1	22.15	0.90	0.35
Mode/Risk	2.37	1	2.37	0.10	0.76
Exp/Style	41.06	1	41.06	1.68	0.20
Exp/Risk	0.02	1	0.02	0.00	0.98
Style/Risk	6.47	1	6.47	0.26	0.61
Explained	221.78	10	22.18	0.90	0.35
Residual	872.64	39	22.37		
Total	1096.42	49	22.38		

Table 6. ANOVA - Error rate

which originally were arbitrarily selected). This seemed to indicate that in unfamiliar situations (where the user was uncertain about exactly what to do next), he was likely to accept default values rather than make his own decisions. To investigate whether any particular user types were more likely to accept these opening defaults, an analysis of variance was performed. As indicated in table 7, there were no significant sources of variance.

Hypothesis 10 - Setting the default response for questions (about the user's desire to see various reports) to 'yes' rather than 'no' will not influence the participant's actual response - was the next to be tested. To do so, ANOVAs were performed for three variables: use of History reports, use of Ordered History reports, and use of Graphs. The assumption was that players with 'yes' defaults would look at

SOURCE VAR.	S.SQ.	DF.	M.SQ.	F	SIGNIF.
Mode	0.63	1	0.63	0.70	0.41
Exp	0.34	1	0.34	0.38	0.54
Style	0.65	1	0.65	0.72	0.40
Risk	0.07	1	0.07	0.08	0.78
Mode/Exp	1.59	1	1.59	1.78	0.19
Mode/Style	0.30	1	0.30	0.33	0.57
Mode/Risk	0.11	1	0.11	0.12	0.73
Exp/Style	0.51	1	0.51	0.57	0.46
Exp/Risk	0.54	1	0.54	0.61	0.44
Style/Risk	0.52	1	0.52	0.58	0.45
Explained	5.18	10	0.52	0.59	0.75
Residual	34.50	39	0.90		
Total	39.68	49	0.81		

Table 7. ANOVA - Opening Defaults

more reports than players with 'no' defaults; hence, Default-value (1='yes', 2='no') was one of the independent variables in the three ANOVAs. To conserve space, the SPSS results are not provided, but in all three cases Default-value was found to be a very insignificant source of variance (ranging from level 0.47 to level 0.97). The implication was that in familiar circumstances (where the user was quite sure of what to do next), default values had no influence upon the user's decisions.

The analysis for hypothesis 11 - Exp, Style, and Risk will all affect whether users accept default values - appears in table 8. It can be seen that cognitive style turned out to be very significant, while neither experience nor risk attitude had any affect. Surprisingly, on average, high analytics made the least use of the default values; in fact, out of every 100 periods they avoided 58 defaults which they could have accepted, while low analytics avoided only about 3. This is a very difficult result to explain, and could certainly use further investigation.

SOURCE VAR.	S.SQ.	DF.	M.SQ.	F	SIGNIF.
Exp	261.02	1	261.02	0.06	0.81
Style	19007.54	1	19007.54	4.38	0.05**
Risk	6805.41	1	6805.41	1.57	0.23
Exp/Style	143.20	1	143.20	0.03	0.86
Exp/Risk	4515.86	1	4515.86	1.04	0.32
Style/Risk	8428.04	1	8428.04	1.94	0.18
Explained	38531.88	6	6421.98	1.48	0.24
Residual	73718.50	17	4336.38		
Total	112250.38	23	4880.45		

Table 8. ANOVA - Acceptance of Defaults

Analysis of variance was also used to test hypothesis 12 - Mode, Exp, Style, and Risk will all affect the extent to which users abbreviate commands. The results (table 9) showed game version and risk attitude both to be very significant. The significance of the game version factor could be explained by the physical difference between the two versions. The risk attitude factor was more interesting: risk-averters abbreviated to significantly less extent than risk-takers (on average, typing 55 of every 100 characters possible, versus 35 of every 100 characters for risk-takers). This may indicate a fear of trying a feature they do not understand, or a mistrust of the computer to interpret their abbreviations correctly.

The last hypothesis regarding program features was

SOURCE VAR.	S.SQ.	DF.	M.SQ.	F	SIGNIF.
Mode	6726.46	1	6726.46	7.49	0.01**
Exp	642.78	1	642.78	0.72	0.40
Style	665.26	1	665.26	0.74	0.40
Risk	3898.38	1	3898.38	4.34	0.04**
Mode/Exp	1062.67	1	1062.67	1.18	0.28
Mode/Style	336.99	1	336.99	0.38	0.54
Mode/Risk	512.94	1	512.94	0.57	0.45
Exp/Style	26.20	1	26.20	0.03	0.86
Exp/Risk	876.26	1	876.26	0.98	0.33
Style/Risk	33.01	1	33.01	0.04	0.85
Explained	16453.66	10	1645.37	1.83	0.09
Residual	35033.00	39	898.28		
Total	51486.66	49	1050.75		

Table 9. ANOVA - Extent of Abbreviation

hypothesis 13 - The length of commands will be the main factor affecting the extent to which they are abbreviated by users. In the analysis, the main effect Length was a two-level variable indicating whether the game with short (3 to 5 letter mnemonics) or long (5 to 8 letter) commands was being played. The analysis appears in table 10 and verifies the hypothesis. Although experience and experience/cognitive style are relatively significant sources of variance, length of commands was clearly the dominating factor. Conclusion: if commands are short, users will tend to type them in full; if long, users will devise abbreviations.

Hypotheses about Comparisons over Time

These seven hypotheses were all related to user learning affects. Each compared user behaviour over the first 10 periods to behaviour over all remaining periods, and was

SOURCE VAR.	S.SQ.	DF.	M.SQ.	F	SIGNIF.
Exp	1381.94	1	1381.94	3.55	0.08**
Style	72.14	1	72.14	0.18	0.67
Risk	16.93	1	16.93	0.04	0.84
Length	8144.06	1	8144.06	20.94	0.00**
Exp/Style	1359.23	1	1359.23	3.49	0.08**
Exp/Risk	606.73	1	606.73	1.56	0.23
Exp/Length	94.17	1	94.17	0.24	0.63
Style/Risk	1091.69	1	1091.69	2.81	0.12*
Style/Length	102.56	1	102.56	0.26	0.62
Risk/Length	261.68	1	261.68	0.67	0.43
Explained	17828.54	10	1782.85	4.45	0.01
Residual	5927.01	15	395.13		
Total	23755.55	25	950.22		

Table 10. ANOVA - Abbreviation by Length

tested by a paired t-test (shown in table 11).

VARIABLE	GROUPING	#	MEAN	STDEV.	T	PROB
Min/Per.	10 Periods	45	0.70	0.21		
	Remainder	45	0.53	0.16	6.07	0.00**
Confid.	10 Periods	45	47.16	24.98		
	Remainder	45	47.29	27.43	-0.03	0.49
Usability	10 Periods	45	5.24	2.35		
	Remainder	45	5.71	2.86	-1.38	0.09**
Abbrev.	10 Periods	45	49.34	33.55		
	Remainder	45	46.42	33.04	2.36	0.01**
Histories	10 Periods	45	14.20	14.40		
	Remainder	45	7.38	10.28	3.26	0.00**
Ord-Hist.	10 Periods	45	9.58	11.06		
	Remainder	45	8.78	11.76	0.38	0.35
Graphs	10 Periods	45	19.16	22.63		
	Remainder	45	25.80	24.64	-2.11	0.02**

Table 11. T-TESTS - Comparisons over Time

Hypothesis 14 - Average time spent playing each period will decrease with time - was clearly supported; user speed increased from 0.70 minutes/period to 0.53 minutes/period (an obvious, yet still encouraging, result).

On the other hand, hypothesis 15 - User confidence will increase with time - was definitely rejected; there was essentially no change in user confidence over time. Apparently, no matter how close they came to the optimum, the users still felt that everyone else must be at the same stage. It may be desirable to provide the user with some indication of comparative performance (reinforcement) whenever possible.

Also supported was hypothesis 16 - User ratings of the usability of the computer program will improve with time. As indicated in table 11, their average ratings changed from 5.24 to 5.71 (on a scale from 1 to 9), indicating some higher appreciation of the program once they had a chance to try many of its features.

Hypothesis 17 - The extent of abbreviation (by unstructured game version players) will increase with time - was also verified, though less dramatically. During the first ten periods, 49 of each 100 characters were typed; during the remainder of the game, 46 were typed. It would seem that people either read in the instructions that they could abbreviate and did so from the start of the game, or they did not abbreviate from the start and only a few learned to do so.

The test of hypothesis 18 - Usage of History reports will decrease with time - was highly significant. The average number of Histories requested per 100 periods dropped from 14 to 7, presumably as people learned the value of the Graphs.

Hypothesis 19 - Usage of Ordered History reports will decrease with time - was rejected. Their use remained quite constant throughout the game; in fact, they were never very popular.

Finally, hypothesis 20 - Usage of Graphs will increase with time - was supported by the results. Initially, only 19 graphs were requested per 100 periods; after 10 periods, though, nearly 26 were requested. It would appear that users quickly learned the value of a more pictorial report. It should also be mentioned that further data analysis revealed that heuristics (low analytics) were the only users who showed no significant increase in their use of Graphs.

Hypotheses about Report Usage and Solution Protocols

The next three hypotheses all relate to usage of reports; in all of these, the dependent variable is the number of reports looked at per 100 periods. The two hypotheses following these both relate to solution protocols.

The first hypothesis in this area is hypothesis 21 - Mode, Exp, Style, and Risk will all affect the use of History reports. The analysis of variance appears in table 12, and shows only game version and experience as significant factors. Users of the structured version used Histories most often (presumably because they are constantly reminded of their existence), while experienced players used them least often (preferring the more informative graphical report). The report frequency by user type was: 18 reports for structured game players and 6 for unstructured; 16 reports for experienced players and 7 for novices.

SOURCE VAR.	S.SQ.	DF.	M.SQ.	F	SIGNIF.
Mode	1639.96	1	1639.96	10.17	0.00**
Exp	808.12	1	808.12	5.01	0.03**
Style	1.82	1	1.82	0.01	0.92
Risk	35.74	1	35.74	0.22	0.64
Mode/Exp	754.70	1	754.70	4.68	0.04**
Mode/Style	71.52	1	71.52	0.44	0.51
Mode/Risk	1.03	1	1.03	0.01	0.94
Exp/Style	13.13	1	13.13	0.08	0.78
Exp/Risk	1.44	1	1.44	0.01	0.92
Style/Risk	2.60	1	2.60	0.02	0.90
Explained	3583.96	10	358.40	2.22	0.04
Residual	6287.26	39	161.21		
Total	9871.22	49	201.45		

Table 12. ANOVA - Use of History Reports

Hypothesis 22 - Mode, Exp, Style, and Risk will all affect the use of Ordered History reports - is analyzed in table 13. The only significant factor is game version (together with an experience interaction). Again, it would seem that structured version users, faced with repeated reminders of the report's existence, select Ordered Histories more often than unstructured game users. No other single factor had much impact (recall from the analysis of hypothesis 19 that this report was not very popular in general).

The test of Hypothesis 23 - Mode, Exp, Style, and Risk will all affect the use of Graphs - had a particularly intriguing result. There were only weakly significant sources

SOURCE VAR.	S.SQ.	DF.	M.SQ.	F	SIGNIF.
Mode	634.87	1	634.87	8.22	0.01**
Exp	121.95	1	121.95	1.58	0.22
Style	29.10	1	29.10	0.38	0.54
Risk	63.46	1	63.46	0.82	0.37
Mode/Exp	343.67	1	343.67	4.45	0.04**
Mode/Style	3.64	1	3.64	0.05	0.83
Mode/Risk	32.88	1	32.88	0.43	0.52
Exp/Style	37.64	1	37.64	0.49	0.49
Exp/Risk	7.47	1	7.47	0.10	0.76
Style/Risk	125.66	1	125.66	1.63	0.21
Explained	1618.67	10	161.87	2.10	0.05
Residual	3013.31	39	77.26		
Total	4631.98	49	94.53		

Table 13. ANOVA - Use of Ordered History Reports

of variance: experience and cognitive style (see table 14). Experienced players requested more Graphs than novices (32 vs. 20) and high analytics requested more than low analtyics (31 vs. 21).

SOURCE VAR.	S.SQ.	DF.	M.SQ.	F	SIGNIF.
Mode	556.49	1	556.49	1.05	0.31
Exp	1393.66	1	1393.66	2.62	0.11*
Style	1170.53	1	1170.53	2.20	0.15*
Risk	268.33	1	268.33	0.51	0.48
Mode/Exp	0.02	1	0.02	0.00	0.99
Mode/Style	258.70	1	258.70	0.49	0.49
Mode/Risk	45.41	1	45.41	0.08	0.77
Exp/Style	59.37	1	59.37	0.11	0.74
Exp/Risk	674.95	1	674.95	1.27	0.27
Style/Risk	852.69	1	852.69	1.60	0.21
Explained	5328.19	10	5328.19	1.00	0.46
Residual	20734.72	39	531.66		
Total	26062.91	49	531.90		

Table 14. ANOVA - Use of Graphs

The last two hypotheses of this thesis relate to users' solution protocols. By plotting all of the <price, quantity> pairs in the order in which they were simulated, a picture of each user's protocol was obtained (see Appendix F for examples). By then "connecting the dots," one could get a good idea of what the original participant was attempting to do. Some participants displayed highly systematic activity, employing a binary search, a gradient search (i.e. hill climbing), a spiralling path, or other explicit model. Other participants used a structured trial and error; they routinely tested every point in the problem space (but with no apparent desire to zoom in on the optimum when neared). Finally, some participants showed no method at all; they just wandered randomly through the problem space.

To test hypothesis 24 - Exp, Style, and Risk will all affect whether users displayed a structured approach to solving the problem (with the emphasis on Style) - the protocol diagram for each user was traced manually, and the approach classified as systematic or not. An ANOVA was then performed, yielding the results in table 15. Although experience was weakly significant, cognitive style was clearly the most significant factor. Comprising 20 of the 27 structured players, high analytics were more frequently systematic and structured, supporting Barrett's classifications (see chapter two).

One other measure was made upon the user protocols: based

SOURCE VAR.	S.SQ.	DF.	M.SQ.	F	SIGNIF.
Exp	1.62	1	1.62	2.34	0.13*
Style	3.31	1	3.31	4.78	0.03**
Risk	0.01	1	0.01	0.01	0.94
Exp/Style	2.61	1	2.61	3.76	0.06**
Exp/Risk	0.44	1	0.44	0.64	0.43
Style/Risk	0.16	1	0.16	0.24	0.63
Explained	9.31	6	1.55	2.24	0.06**
Residual	29.81	43	0.69		
Total	39.12	49	0.80		

Table 15. ANOVA - Protocol Structure

SOURCE VAR.	S.SQ.	DF.	M.SQ.	F	SIGNIF.
Exp	0.79	1	0.79	3.36	0.07**
Style	0.08	1	0.08	0.32	0.57
Risk	0.03	1	0.03	0.14	0.72
Exp/Style	0.27	1	0.27	1.13	0.29
Exp/Risk	0.75	1	0.75	3.18	0.08**
Style/Risk	0.02	1	0.02	0.08	0.77
Explained	2.15	6	0.36	1.51	0.20
Residual	10.17	43	0.24		
Total	12.32	49	0.25		

Table 16. ANOVA - Protocol Dispersion

upon the extent to which participants searched the entire problem space, or just concentrated upon one small area, the protocols were manually classified as dispersed or not. Then hypothesis 25 - Exp, Style, and Risk will all affect the amount of dispersion displayed in their search for the optimum (with the emphasis again on Style) - was tested. As indicated in table 16, the only significant factor was experience (together with a risk attitude interaction); apparently, experienced players were more familiar with this type of task and did not find any need to "feel around" the entire problem space. Neither of the psychological variables could explain much of this behaviour.

As mentioned in the previous chapter, a summary of these results appears in appendix G.

Chapter Seven

CONCLUSIONS

In this thesis, a new research tool (in the form of an interactive computer program) has been introduced. The motivation for this has been described: to present an example of a convenient, "idiot-proof" computer program, and to facilitate investigation of some aspects of man-machine communication which could be of interest to other information systems researchers.

Some of the related literature has been discussed; then the user engineering of the computer program was described in detail. Next, the actual process of data collection for this research was presented. The pre-testing for this research was described and shown to be quite convenient for both administrator and participant, taking just over one half-hour. The details of the computer experiment were then presented. Again, the convenience aspect could not be over-emphasized: the game lasted only one half-hour, making it easy to administer and minimizing the possibility of subjects getting bored or needing to hurry to get it over with.

In the actual running of the experiments, two items are particularly noteworthy. First, despite heavy emphasis on the need to carefully read the instructions in advance, it was found that some people just did not do it. This suggests a need to personally tutor every new user of a computer system or otherwise reiterate the instructions (perhaps on the terminal screen); no matter how well the documentation may be written, some people just will not read it or take the time to properly understand it - and the results can be disastrous (poor results now and lack of faith in computers in the future).

The second problem observed while administering the game involved getting novice users started with the unstructured version of the game; the concept of a general command processor appeared to be just too sophisticated for them. Personal attention was needed to explain the task and sometimes demonstrate it. It seemed quite clear that novices would be happier with the structured game version - at least until they understood what was happening. To this end, recall from chapter three that the extra programming necessary to write a program which could be run in either mode was quite minimal.

Finally, the process of data capture and conversion was presented. It was mentioned that data about user performance, behaviour, attitude, and even solution protocol were all collected by the computer program. This data was then analyzed with SPSS, using analysis of variance and t-tests (both normal-one-tailed and paired). The results of these analyses are now reviewed, this time in a different order and with added discussion.

In relating psychological variables to performance, it was found that cognitive style had a strong effect upon whether people finished the game on time, and upon their confidence level throughout the game. High analytics finished more often and were more confident than low analytics, indicating that this game (and perhaps many mathematical tasks?) may favor high analytics. Cognitive style also impacted report usage: it was found that heuristics (low analytics) were the only group which neither decreased its use of history reports nor increased its dependence upon graphical reports later in the game (implying a preference for the less structured and less summarized feedback). The high analytics displayed a significant tendency to avoid accepting default responses (a result requiring further investigation). Finally, in the analysis of user solution protocols, it was found that low analytics were significantly less structured than high analytics in their search for the optimum (supporting the model suggested by Barrett).

Another psychological variable, risk attitude, was found to significantly affect playing speed and confidence; risk-takers spent less time per move and were more confident than risk-averters (questionning the findings of Taylor and Dunnette). Also, risk-averters were found to abbreviate commands to much less extent than other users. If this is caused by a mistrust of the computer, efforts should be made to dispel this fear.

However, the most dominant factor on all dimensions was the user's previous experience with on-line computer systems. Experienced players were much faster, finished more often, and were significantly more confident than novices. This clearly indicates the importance of explicitly recognizing these factors in any computer research (and probably any research, for that matter). Experienced users were happy with either game version, while, as noted earlier, novices were initially lost with the unstructured game. Finally, experienced participants made the least use of History reports, and showed the least amount of dispersion in their solution protocols, both indicating an ability to detect and disregard less relevant material. Again, it is pointed out that experienced subjects seem to have an advantage in computerized research.

Game version, on the other hand, was found (statistically) to be a very weak factor, affecting neither speed, termination, nor confidence. It would seem that, for reasonably simple tasks, both versions are equally useful. The ideal, therefore, would be to provide the user with both alternatives, and let him choose whichever is more comfortable for him (a choice which may change with time). However, it should be noted that game version did affect some behaviour: users of the structured version (which reminds players of the availability of the reports each period) requested significantly more History and Ordered History reports.

Another area which was considered briefly was error rate. No differences were found among user types with respect to making typographical or range errors. In fact, there were very few errors made by any participants in this game; the user engineering aspects of the computer program (minimized memorization, indication of allowed responses, unlimited abbreviation, etc.) seem to have minimized the possibilities for error.

The impact of defaults under various circumstances was also studied. It was found that the opening price and quantity defaults were accepted by over one-half of the participants in the first period of the game, whereas setting the default response to questions about the user's desire to see a report to 'yes' rather than 'no' had no significant effect upon whether they actually requested that report. Hence, in less well-defined situations, people appear to select the default value rather than think for themselves. It is suggested that default values not be provided in these circumstances (as they may bias the results). On the other hand, in situations where the choice is clear, defaults appear not to influence the user, and are recommended as an aid to him (to minimize unnecessary typing).

The last area examined concerning program usage was the effect of command length upon the extent of abbreviation by users. Players of the game with 3 to 5 letter mnemonic commands abbreviated far less frequently than players of the game with 5 to 8 letter commands. There are two possible implications of this: to "force" users to abbreviate commands (and presumably play faster), intentionally make the commands long; to "force" users to remember the commands in full, make them reasonably short (but do not compromise their intelligibility). It is proposed that the former is more appropriate when there are only a few commands and the latter is best when the number of commands is quite large.

The last area considered in this research compared user performance and behaviour in the first 10 periods of the game to the remainder of the game. The analysis of results revealed that playing speed, extent of abbreviation, and use of Graphs all increased significantly over time, and use of History reports decreased over time (all of which were desirable from a systems designer's point of view). User ratings of the usability of the program also increased slightly, indicating that users appreciated the program more after they had time to get comfortable with it. A surprising result was that user confidence did not change with time; it would seem that no matter how well people are doing, if they do not know how everyone else is doing, they assume that they are performing only average. Perhaps some comparative performance feedback could remedy this (when it is available, of course).

In summary, a very simple, yet effective (and enjoyable!) research tool has been described, and the results of an experiment using it have been presented. Some of the findings of previous researchers have been confirmed; some new results have been provided about the man-machine interface. Clearly, many of these results have touched only the surface, and much more research could be done in this area. For example: the use of reports could be investigated further by having versions with and without graphical reports and studying the impact upon performance, behaviour, attitude, and solution protocol; the comparisons over time could be applied tc the period-by-period time series data, rather than to two averages (first 10 periods versus remainder); the solution protccols could be studied more carefully and scientifically; or other aspects of special program features could be investigated, including utilization of typeahead capabilities and its impact upcn playing speed, error rate, etc.

FOOTNOTES

[1] DeGreene, Kenyon B. "Man-Computer Interrelationships," System Psychology, Kenyon B. DeGreene (ed.), (New York: McGraw-Hill Book Company), 1970, pp. 281-336.

[2] ibid., p. 282.

[3] Keen, Peter G. W. "The Implications of Cognitive Style for Individual Decision Making," D.B.A. Thesis, Harvard University (1973), Part III, Ch. 10, pp.28-31.

[4] Botkin, J. W. "An Intuitive Computer System: A Cognitive Approach to the Management Learning Process," D.B.A. Thesis, Harvard University (1973).

[5] Benbasat, Izak, and Roger Schroeder. "An Experimental Investigation of Some MIS Design Variables," MIS Quarterly, V. 1, No. 1 (March, 1977), pp. 37-49.

[6] ibid.

[7] Benbasat, Izak, and Ronald N. Taylor. "The Impact of Cognitive Styles on Information System Design," Working Paper No. 518, Faculty of Commerce and Business Administration, University of British Columbia.

[8] Benbasat, Izak, and Albert S. Dexter. "Value and Events Approaches to Accounting: An Experimental Evaluation," Working Paper No. 488, Faculty of Commerce and Business Administration, University of British Columbia (1978).

[9] Ferguson, Robert L., and Curtis H. Jones. "A Computer Aided Decision System," Management Science, V. 15, No. 10 (June, 1969), pp. B-550 - B-561.

[10] Wasserman, Anthony I. "The Design of 'Idiot-Proof' Interactive Programs," Proceedings, 1973 National Computer Conference, V. 42, pp. M34-M38.

[11] ibid., p. M35.

[12] ibid., p. M38.

[13] Hansen, Wilfred J. "User Engineering Principles for Interactive Systems," Proceedings, 1971 AFIPS Fall Joint Computer Conference, pp. 523-532.

[14] ibid., p. 528.

[15] Eason, K. D. "The Manager as a Computer User," Applied Ergonomics, V. 5, No. 1 (1974), pp. 9-14.

[16] ibid., p. 14.

[17] Barrett, Gerald V., Carl L. Thornton, and Patrick A. Cabe. "Human Factors Evaluation of a Computer Based Information Storage and Retrieval System," Human Factors, V. 10, No. 4 (August, 1968), pp. 431-436.

[18] ibid., p. 435.

[19] Mock, Theodore J. "A Longitudinal Study of Some Information Structure Alternatives," Data Base, V. 5, No. 2,3 & 4 (Winter, 1973), pp. 40-44.

[20] ibid., p. 45.

[21] Wynne, Bayard E. And Gary W. Dickson. "Experienced Managers' Performance in Experimental Man-Machine Decision System Simulation," Academy of Management Journal, V. 18, No. 1 (March, 1975), pp. 25-40.

[22] ibid., p. 39.

[23] op. cit., Benbasat and Taylor.

[24] ibid., pp. 14-15.

[25] Taylor, Ronald N., and Marvin D. Dunnette. "Relative Contribution of Decision-Maker Attributes to Decision Processes," Organizational Behavior and Human Performance, V. 12, No. 2 (October, 1974).

[26] ibid., p. 296.

[27] Taylor, Ronald N. "Psychological Determinants of Bounded Rationality: Implications for Decision-Making Strategies," Decision Sciences, V. 6, No. 3 (July, 1975), pp. 409-429.

[28] Davis, Gordon B. Management Information Systems: Conceptual nt, (New York: McGraw-Hill Book Company), 1974, pp. 150-151.

[29] Meadow, Charles T. Man-Machine Communication, (New York: Wiley-Interscience), 1970.

[30] Martin, James. Design of Man-Computer Dialogues, (Englewood Cliffs, New Jersey: Prentice-Hall, Inc.), 1973.

[31] ibid., pp. 87-88.

[32] Witkin, H. A., P. K. Oltman, E. Ruskin, and S. A. Karp. _The Embedded Figures Test_, (Palo Alto, California: Consulting Psychologist Press, Inc.), 1971.

[33] Kogan, N. And M. A. Wallach. _Risk Taking: A Study in Cognition and Personality_, (New York: Holt, Rinehart, and Winston), 1964.

[34] Nie, N. H., C. H. Hull, J. G. Jenkins, K. Steinbrenner, and D. H. Bent. _Statistical Package for the Social Sciences_, Second Edition, (New York: McGraw-Hill Book Company), 1975.

[35] MacCrimmon, Kenneth R. "An Experimental Study of the Decision Making Behavior of Business Executives," Ph.D. Thesis, University of California, Los Angelas (1965).

[36] ibid., p. 207.

BIBLIOGRAPHY

Barrett, Gerald V., Carl L. Thornton, and Patrick A. Cabe. "Human Factors Evaluation of a Computer Based Information Storage and Retrieval System," Human Factors, V. 10, No. 4 (August, 1968), pp. 431-436.

Benbasat, Izak, and Albert S. Dexter. "Value and Events Approaches to Accounting: An Experimental Evaluation," Working Paper No. 488, Faculty of Commerce and Business Administration, University of British Columbia (1978).

Benbasat, Izak, and Roger Schroeder. "An Experimental Investigation of Some MIS Design Variables," MIS Quarterly, V. 1, No. 1 (March, 1977), pp. 37-49.

Benbasat, Izak, and Ronald N. Taylor. "The Impact of Cognitive Styles on Information System Design," Working Paper No. 518, Faculty of Commerce and Business Administration, University of British Columbia.

Botkin, J. W. "An Intuitive Computer System: A Cognitive Approach to the Management Learning Process," D.B.A. Thesis, Harvard University (1973).

Davis, Gordon B. Management Information Systems: Conceptual Foundations, Structure, and Development, New York: McGraw-Hill Book Company, 1974.

DeGreene, Kenyon B. "Man-Computer Interrelationships," System Psychology, Kenyon B. DeGreene (ed.), New York: McGraw-Hill Book Company, 1970, pp. 281-336.

Eason, K. D. "The Manager as a Computer User," Applied Ergonomics, V. 5, No. 1 (1974), pp. 9-14.

Ferguson, Robert L., and Curtis H. Jones. "A Computer Aided Decision System," Management Science, V. 15, No. 10 (June, 1969), pp. B-550 - B-561.

Hansen, Wilfred J. "User Engineering Principles for Interactive Systems," Proceedings, 1971 AFIPS Fall Joint Computer Conference, pp. 523-532.

Keen, Peter G. W. "The Implications of Cognitive Style for Individual Decision Making," D.B.A. Thesis, Harvard University (1973).

Kogan, N. and M. A. Wallach. Risk Taking: A Study in Cognition and Personality, New York: Holt, Rinehart, and Winston, 1964.

MacCrimmon, Kenneth R. "An Experimental Study of the Decision Making Behavior of Business Executives," Ph.D. Thesis, University of California, Los Angelas (1965).

Martin, James. Design of Man-Computer Dialogues, Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1973.

Meadow, Charles T. Man-Machine Communication, New York: Wiley-Interscience, 1970.

Mock, Theodore J. "A Longitudinal Study of Some Information Structure Alternatives," Data Base, V. 5, No. 2,3 & 4 (Winter, 1973), pp. 40-44.

Nie, N. H., C. H. Hull, J. G. Jenkins, K. Steinbrenner, and D. H. Bent. Statistical Package for the Social Sciences, Second Edition, New York: McGraw-Hill Book Company, 1975.

Taylor, Ronald N. "Psychological Determinants of Bounded Rationality: Implications for Decision-Making Strategies," Decision Sciences, V. 6, No. 3 (July, 1975), pp. 409-429.

Taylor, Ronald N., and Marvin D. Dunnette. "Relative Contribution of Decision-Maker Attributes to Decision Processes," Organizational Behavior and Human Performance, V. 12, No. 2 (October, 1974), pp. 286-298.

Wasserman, Anthony I. "The Design of 'Idiot-Proof' Interactive Programs," Proceedings, 1973 National Computer Conference, V. 42, pp. M34-M38.

Witkin, H. A., P. K. Oltman, E. Ruskin, and S. A. Karp. The Embedded Figures Test, Palo Alto, California: Consulting Psychologist Press, Inc., 1971.

Wynne, Bayard E. and Gary W. Dickson. "Experienced Managers' Performance in Experimental Man-Machine Decision System Simulation," Academy of Management Journal, V. 18, No. 1 (March, 1975), pp. 25-40.

Appendix A

PROGRAM LISTING

A listing of the source code for the computer game appears on the next 21 pages. The program is written entirely in FORTRAN and is about 1000 lines long. It uses some subprograms (for timing, file control, and character comparison) which are specific to the University of British Columbia, and hence can probably only serve as an example for others. In the pages to follow, the program comments should suffice as general documentaticn.

```
C       CRT GAME FOR THESIS DATA COLLECTION
C       P. MASULIS    -DECEMBER,1977
C
C       TO RUN THIS GAME:
C           R *FTN SCARDS=PSM.FTN SPUNCH=PSM
C           R PSM+CPU:LIB PAR=CCC,YYY,LLL,<USER'S NAME>
C              WHERE CCC=CYC *   FOR STRUCTURED INPUT
C                       =CMD     FOR UNSTRUCTURED INPUT
C                    YYY=YES *   FOR 'YES' DEFAULT/LONG COMMANDS
C                       =NO      FOR 'NO' DEFAULT/SHORT COMMANDS
C                    LLL=LOW *   FOR (10,25) INITIAL (PRICE,QTY)
C                       =HI      FOR (20,45) INITIAL (PRICE,QTY)
C                            * INDICATES DEFAULT VALUE
C
C
        IMPLICIT INTEGER(A-Z)
        LOGICAL EQUC
        INTEGER*2 MODE(30),CMD/'CM'/,NO/'NO'/,HI/'HI'/,RECOVR/'RE'/
        LOGICAL*1 YNDEF,Y/'Y'/,E/'E'/,S/'S'/,N/'N'/,O/'O'/,BLANK/' '/
        LOGICAL*1 SDUM(10)
        INTEGER*2 NAME(6)
        EQUIVALENCE(NAME(1),MODE(9))
        INTEGER*2 DATA,SAVE
        LOGICAL ATTN
        REAL RZ
        COMMON PERIOD,PRICE,QTY,PROFIT,DATA(30,70),SAVE(100,4),ATTN
        COMMON MINPRF,MAXPRF,MAXPTR,YNDEF(4),REP(3),NUMNUM,NUMHLP
        COMMON NUMLIT,NUMGET,NUMCMD,NUMDEF,NUMERR,MAXCHR,NUMCHR,RZ
        COMMON NUMNDA,NUMNDN
C
C       PRELOAD COMMON VARIABLES
C
        NUMLIT=0
        NUMGET=0
        NUMCMD=0
        NUMDEF=0
        NUMERR=0
        MAXCHR=0
        NUMCHR=0
        NUMNUM=0
        NUMHLP=0
        NUMNDA=0
        NUMNDN=0
        REP(1)=0
        REP(2)=0
        REP(3)=0
        MAXPTR=1
        PERIOD=0
        MAXPRF=0
        MINPRF=999
        SAVE(MAXPTR,4)=0
C
C       SETUP I/O
C
        CALL ATNTRP(ATTN)
        LEN=7
        CALL CNTRL('RATE 10',LEN,6)
```

```
      LEN=4
      CALL CNTRL('ROLL',LEN,6)
      CALL FTNCMD('DEFAULT 7=PSM#1 ',16)
      CALL FTNCMD('DEFAULT 8=IZAK:FUNCTION ',24)
C
C     CHECK FOR RECOVERY RUN
C
      CALL PAR(MODE(3),NI,24,&6,&6)
    6 IF(MODE(3).NE.RECOVR) GOTO 19
      READ(7,7) MX,MY,RZ
    7 FORMAT(24X,I3,5X,I3,5X,F5.3)
      CALL READPF(NAME,MX,MY,RZ)
      READ(7,8) MODE(3),(YNDEF(I),I=1,3)
    8 FORMAT(10X,A2,26X,3A1)
   11 READ(7,12,END=60) ICODE,PRICE,QTY,MTIM
   12 FORMAT(I2,3X,2I3,29X,I5)
      IF(ICODE.EQ.1) CALL SIMUL(.TRUE.,MTIM)
      GOTO 11
C
C     CREATE FILES IF NOT RECOVERY RUN
C
   19 CALL DESTRY('PSM#1 ')
      CALL CREATE('PSM#1 ',1,0,256)
      CALL OUTMES(1)
C
C     SELECT APPROPRIATE MODES
C
   30 IF(MODE(5).EQ.NO) GOTO 35
      YNDEF(1)=Y
      YNDEF(2)=E
      YNDEF(3)=S
      GOTO 40
   35 YNDEF(1)=N
      YNDEF(2)=O
      YNDEF(3)=BLANK
   40 IF(MODE(7).EQ.HI) GOTO 50
      PRICE=10
      QTY=25
      GOTO 51
   50 PRICE=20
      QTY=45
C
C     READ IN PROFIT FUNCTION, THEN CALL PROPER INPUT MONITOR
C
   51 MX=0
      CALL READPF(NAME,MX,MY,RZ)
   60 IF(MODE(3).EQ.CMD) CALL GCMAND
      CALL GCYCLE
      STOP
      END
```

```
      SUBROUTINE GCMAND
C
C     INPUT MONITOR - UNSTRUCTURED INPUT
C
      IMPLICIT INTEGER(A-Z)
      LOGICAL EQUC
      LOGICAL*1 DUM(2),CMD(10),N/'N'/,LNGCMD/.TRUE./
      INTEGER*2 CMD2,G/' G'/,J/' J'/,S/' S'/,NULL/'  '/
      EQUIVALENCE(DUM(2),CMD(1))
      EQUIVALENCE(DUM(1),CMD2)
      LOGICAL*1 YNDEF
      INTEGER*2 DATA,SAVE
      LOGICAL ATTN
      REAL RZ
      COMMON PERIOD,PRICE,QTY,PROFIT,DATA(30,70),SAVE(100,4),ATTN
      COMMON MINPRF,MAXPRF,MAXPTR,YNDEF(4),REP(3),NUMNUM,NUMHLP
      COMMON NUMLIT,NUMGET,NUMCMD,NUMDEF,NUMERR,MAXCHR,NUMCHR,RZ
      COMMON NUMNDA,NUMNDN
C
C     INITIALIZATION
C
      CMD2=NULL
      CALL TIME(0)
      IF(EQUC(YNDEF(1),N))  LNGCMD=.FALSE.
      IF(PERIOD.GT.0) GOTO 10
      CALL GETLIT(.TRUE.,CMD,LEN,10)
      WRITE(7,5) PRICE,QTY,(YNDEF(I),I=1,3)
    5 FORMAT(' OMODE=2 (CMD)  PRICE=',I2,'  QTY=',I2,'  DEF=',3A1)
      CALL OUTMES(2)
      GOTO 80
C
C     READ AND PROCESS COMMAND
C
   10 CALL GETLIT(.FALSE.,CMD,LEN,19)
      IF (LEN.EQ.0) GOTO 80
      IF(CMD2.LT.S) GOTO 12
      RTN=CMD2-S+13
      GOTO 18
   12 IF(CMD2.LT.J) GOTO 15
      RTN=CMD2-J+4
      GOTO 18
   15 RTN=CMD2-G+1
   18 IF(RTN.LT.1 .OR. RTN.GT.13) GOTO 80
      GOTO(70,50,80,80,80,80,80,80,60,20,30,80,40), RTN
C
C     SET PRICE
C
   20 CALL GETNUM(.FALSE.,PRICE,1,30,11,12)
      MAXCHR=MAXCHR+5
      NUMCHR=NUMCHR+LEN
      NUMCMD=NUMCMD+1
      GOTO 10
C
C     SET QUANTITY
C
   30 CALL GETNUM(.FALSE.,QTY,1,70,13,14)
      MAXCHR=MAXCHR+3
```

```
      IF(LNGCMD) MAXCHR=MAXCHR+5
      NUMCHR=NUMCHR+LEN
      NUMCMD=NUMCMD+1
      GOTO 10
C
C     SIMULATE ANOTHER PERIOD
C
   40 CALL SIMUL(.FALSE.,0)
      MAXCHR=MAXCHR+3
      IF(LNGCMD) MAXCHR=MAXCHR+5
      NUMCHR=NUMCHR+LEN
      NUMCMD=NUMCMD+1
      GOTO 10
C
C     DISPLAY HISTORY REPORT
C
   50 CALL HISTRY
      MAXCHR=MAXCHR+4
      IF(LNGCMD) MAXCHR=MAXCHR+3
      NUMCHR=NUMCHR+LEN
      NUMCMD=NUMCMD+1
      GOTO 10
C
C     DISPLAY SORTED HISTORY
C
   60 CALL SORTH
      MAXCHR=MAXCHR+3
      IF(LNGCMD) MAXCHR=MAXCHR+5
      NUMCHR=NUMCHR+LEN
      NUMCMD=NUMCMD+1
      GOTO 10
C
C     DISPLAY GRAPH
C
   70 CALL SGRAPH
      MAXCHR=MAXCHR+5
      NUMCHR=NUMCHR+LEN
      NUMCMD=NUMCMD+1
      GOTO 10
C
C     USER COMMAND ERROR
C
   80 IMES=31
      IF(LNGCMD) IMES=18
      CALL OUTMES(IMES)
      IF(LEN.GT.0) NUMERR=NUMERR+1
      CALL CLRSTR
      GOTO 10
      END
```

```
      SUBROUTINE GCYCLE
C
C     INPUT MONITOR - STRUCTURED INPUT
C
      IMPLICIT INTEGER(A-Z)
      LOGICAL EQUC
      LOGICAL*1 BOOL(10)/10*' '/,Y/'Y'/,N/'N'/
      LOGICAL*1 YNDEF
      INTEGER*2 DATA,SAVE
      LOGICAL ATTN
      REAL RZ
      COMMON PERIOD,PRICE,QTY,PROFIT,DATA(30,70),SAVE(100,4),ATTN
      COMMON MINPRF,MAXPRF,MAXPTR,YNDEF(4),REP(3),NUMNUM,NUMHLP
      COMMON NUMLIT,NUMGET,NUMCMD,NUMDEF,NUMERR,MAXCHR,NUMCHR,RZ
      COMMON NUMNDA,NUMNDN
C
C     INITIALIZATION
C
      CALL TIME(0)
      IF(PERIOD.GT.0) GOTO 10
      CALL GETLIT(.TRUE.,BOOL,LEN,10)
      WRITE(7,5) PRICE,QTY,(YNDEF(I),I=1,3)
    5 FORMAT(' 0MODE=1 (CYC)   PRICE=',I2,'   QTY=',I2,'   DEF=',3A1)
      CALL OUTMES(3)
C
C     GET PRICE & QTY, AND SIMULATE
C
   10 CALL GETNUM(.TRUE.,PRICE,1,30,11,12)
      CALL GETNUM(.TRUE.,QTY,1,70,13,14)
      NUMCMD=NUMCMD+2
      CALL SIMUL(.FALSE.,0)
C
C     DISPLAY HISTORY REPORT - IF DESIRED
C
   20 CALL GETLIT(.TRUE.,BOOL,LEN,15)
      IF(EQUC(BOOL(1),N).OR.(LEN.EQ.0 .AND. EQUC(YNDEF(1),N))) GOTO 30
      IF(EQUC(BOOL(1),Y).OR.(LEN.EQ.0 .AND. EQUC(YNDEF(1),Y))) GOTO 25
      CALL OUTMES(8)
      NUMERR=NUMERR+1
      GOTO 20
   25 CALL HISTRY
      IF(LEN.GT.0) MAXCHR=MAXCHR+1
   30 IF(LEN.GT.0) MAXCHR=MAXCHR+2
      IF(EQUC(BOOL(1),YNDEF(1))) NUMNDA=NUMNDA+1
      NUMCHR=NUMCHR+LEN
      IF(LEN.EQ.0) NUMDEF=NUMDEF+1
C
C     DISPLAY SORTED HISTORY - IF DESIRED
C
      CALL GETLIT(.TRUE.,BOOL,LEN,16)
      IF(EQUC(BOOL(1),N).OR.(LEN.EQ.0 .AND. EQUC(YNDEF(1),N))) GOTO 40
      IF(EQUC(BOOL(1),Y).OR.(LEN.EQ.0 .AND. EQUC(YNDEF(1),Y))) GOTO 35
      CALL OUTMES(8)
      NUMERR=NUMERR+1
      GOTO 30
   35 CALL SORTH
      IF(LEN.GT.0) MAXCHR=MAXCHR+1
```

```
   40 IF(LEN.GT.0) MAXCHR=MAXCHR+2
      IF(EQUC(BOOL(1),YNDEF(1))) NUMNDA=NUMNDA+1
      NUMCHR=NUMCHR+LEN
      IF(LEN.EQ.0) NUMDEF=NUMDEF+1
C
C     DISPLAY GRAPH - IF DESIRED
C
      CALL GETLIT(.TRUE.,BOOL,LEN,17)
      IF(EQUC(BOOL(1),N).OR.(LEN.EQ.0 .AND. EQUC(YNDEF(1),N))) GOTO 50
      IF(EQUC(BOOL(1),Y).OR.(LEN.EQ.0 .AND. EQUC(YNDEF(1),Y))) GOTO 45
      CALL OUTMES(8)
      NUMERR=NUMERR+1
      GOTO 40
   45 CALL SGRAPH
      IF(LEN.GT.0) MAXCHR=MAXCHR+1
   50 IF(LEN.GT.0) MAXCHR=MAXCHR+2
      IF(EQUC(BOOL(1),YNDEF(1))) NUMNDA=NUMNDA+1
      NUMCHR=NUMCHR+LEN
      IF(LEN.EQ.0) NUMDEF=NUMDEF+1
      NUMCMD=NUMCMD+3
      GOTO 10
      END
```

```
      SUBROUTINE GETLIN(STRING,LENGTH)
C
C     GET AN INPUT LINE FROM THE CRT
C
      IMPLICIT INTEGER(A-Z)
      LOGICAL EQUC
      LOGICAL*1 STRING(60),BLANK/' '/
      LOGICAL*1 YNDEF
      INTEGER*2 DATA,SAVE
      LOGICAL ATTN
      REAL RZ
      COMMON PERIOD,PRICE,QTY,PROFIT,DATA(30,70),SAVE(100,4),ATTN
      COMMON MINPRF,MAXPRF,MAXPTR,YNDEF(4),REP(3),NUMNUM,NUMHLP
      COMMON NUMLIT,NUMGET,NUMCMD,NUMDEF,NUMERR,MAXCHR,NUMCHR,RZ
      COMMON NUMNDA,NUMNDN
C
C     READ A 60 CHARACTER STRING FROM USER
C
      NUMGET=NUMGET+1
      LENGTH=0
      WRITE(6,10)
   10 FORMAT('&:')
      DO 15 I=1,60
          STRING(I)=BLANK
   15     CONTINUE
      READ(5,20) (STRING(I),I=1,60)
   20 FORMAT(60A1)
C
C     STRIP OFF TRAILING BLANKS
C
      DO 30 I=1,60
          IF(EQUC(STRING(61-I),BLANK)) GOTO 30
          LENGTH=61-I
          GOTO 40
   30     CONTINUE
   40 RETURN
      END
```

```
      SUBROUTINE GETLIT(NEWSTR,LIT,LITLEN,PROMPT)
C
C     GET NEXT LITERAL IN INPUT STRING (UP TO 10 CHARS)
C     LITERAL IS DELIMITED BY SPACES OR A COMMA
C
      IMPLICIT INTEGER(A-Z)
      LOGICAL EQUC,NEWSTR,ATTN
      LOGICAL*1 STRING(60),CHAR,LIT(10),BLANK/' '/,COMMA/','/
      INTEGER SPTR/1/,LENGTH/0/
      LOGICAL*1 YNDEF
      INTEGER*2 DATA,SAVE
      REAL RZ
      COMMON PERIOD,PRICE,QTY,PROFIT,DATA(30,70),SAVE(100,4),ATTN
      COMMON MINPRF,MAXPRF,MAXPTR,YNDEF(4),REP(3),NUMNUM,NUMHLP
      COMMON NUMLIT,NUMGET,NUMCMD,NUMDEF,NUMERR,MAXCHR,NUMCHR,RZ
      COMMON NUMNDA,NUMNDN
C
C     INITIALIZATION
C
      NUMLIT=NUMLIT+1
      LITLEN=0
      DO 5 I=1,10
          LIT(I)=BLANK
    5     CONTINUE
      IF(.NOT.NEWSTR .AND. SPTR.LE.LENGTH) GOTO 10
      CALL OUTMES(PROMPT)
      CALL GETLIN(STRING,LENGTH)
      IF(LENGTH.EQ.0) GOTO 50
      SPTR=1
C
C     STRIP OFF LEADING BLANKS
C
   10 IF(.NOT.EQUC(STRING(SPTR),BLANK)) GOTO 20
      SPTR=SPTR+1
      GOTO 10
C
C     BUILD ACTUAL LITERAL, CHAR BY CHAR
C
   20 CHAR=STRING(SPTR)
      IF(EQUC(CHAR,BLANK) .OR. EQUC(CHAR,COMMA)) GOTO 30
      LITLEN=LITLEN+1
      IF(LITLEN.LE.10) LIT(LITLEN)=CHAR
      SPTR=SPTR+1
      IF(SPTR.GT.LENGTH) GOTO 50
      GOTO 20
C
C     STRIP OFF TRAILING BLANKS AND COMMAS
C
   30 IF(.NOT.EQUC(STRING(SPTR),BLANK)) GOTO 40
      SPTR=SPTR+1
      GOTO 30
   40 IF(EQUC(STRING(SPTR),COMMA)) SPTR=SPTR+1
   50 RETURN
      ENTRY CLRSTR
      SPTR=100
      RETURN
      END
```

```
      SUBROUTINE GETNUM(NEWNUM,NUMBER,LOW,HIGH,HELP,PROMPT)
C
C     GET NEXT INTEGER NUMBER IN INPUT STRING
C
      IMPLICIT INTEGER(A-Z)
      LOGICAL*1 DUM(2),LITNUM(11)
      LOGICAL NEWNUM,BOOL
      INTEGER*2 NMCHR2,ZERO/' 0'/,NULL/'  '/
      EQUIVALENCE(DUM(1),NMCHR2)
      LOGICAL*1 YNDEF
      INTEGER*2 DATA,SAVE
      LOGICAL ATTN
      REAL RZ
      COMMON PERIOD,PRICE,QTY,PROFIT,DATA(30,70),SAVE(100,4),ATTN
      COMMON MINPRF,MAXPRF,MAXPTR,YNDEF(4),REP(3),NUMNUM,NUMHLP
      COMMON NUMLIT,NUMGET,NUMCMD,NUMDEF,NUMERR,MAXCHR,NUMCHR,RZ
      COMMON NUMNDA,NUMNDN
C
C     INITIALIZATION
C
      NUMNUM=NUMNUM+1
      NMCHR2=NULL
      DEFALT=NUMBER
      BOOL=NEWNUM
   10 CALL GETLIT(BOOL,LITNUM,LITLEN,PROMPT)
      IF(LITLEN.EQ.0) GOTO 40
C
C     CONVERT STRING LITERAL TO INTEGER
C
      NUMBER=0
      DO 20 I=1,LITLEN
          DUM(2)=LITNUM(I)
          DIGIT=NMCHR2-ZERO
          IF(DIGIT.LT.0 .OR. DIGIT.GT.9) GOTO 30
          NUMBER=NUMBER*10+DIGIT
   20     CONTINUE
      IF(NUMBER.LT.LOW .OR. NUMBER.GT.HIGH) GOTO 30
      IF(NUMBER.EQ.DEFALT) NUMNDN=NUMNDN+1
      GOTO 50
C
C     REQUEST USER TO RE-INPUT THE NUMBER
C
   30 CALL OUTMES(HELP)
      NUMHLP=NUMHLP+1
      BOOL=.TRUE.
      GOTO 10
C
C     USER TYPED JUST "RETURN", GIVE HIM THE DEFAULT - IF ANY
C
   40 IF(DEFALT.LT.0) GOTO 30
      NUMBER=DEFALT
      NUMDEF=NUMDEF+1
   50 RETURN
      END
```

```
      SUBROUTINE HISTRY
C
C     OUTPUT MOST RECENT GAME RESULTS
C
      IMPLICIT INTEGER(A-Z)
      LOGICAL*1 YNDEF
      INTEGER*2 DATA,SAVE
      LOGICAL ATTN
      REAL RZ
      COMMON PERIOD,PRICE,QTY,PROFIT,DATA(30,70),SAVE(100,4),ATTN
      COMMON MINPRF,MAXPRF,MAXPTR,YNDEF(4),REP(3),NUMNUM,NUMHLP
      COMMON NUMLIT,NUMGET,NUMCMD,NUMDEF,NUMERR,MAXCHR,NUMCHR,RZ
      COMMON NUMNDA,NUMNDN
C
      REP(1)=REP(1)+1
      IF(PERIOD.GT.0) GOTO 5
      CALL OUTMES(7)
      GOTO 30
    5 K=25
      IF(PERIOD.LT.K) K=PERIOD
      CALL OUTMES(28)
      DO 20 I=1,K
          J=PERIOD-K+I
          WRITE(6,10) J,SAVE(J,1),SAVE(J,2),SAVE(J,3)
   10     FORMAT(1X,4I8)
   20     CONTINUE
   30 RETURN
      END
```

```
      SUBROUTINE OUTMES(MSG)
C
C     PRINT A MESSAGE ON THE CRT SCREEN
C
      IMPLICIT INTEGER(A-Z)
      LOGICAL*1 YNDEF
      INTEGER*2 DATA,SAVE
      LOGICAL ATTN
      REAL RZ
      COMMON PERIOD,PRICE,QTY,PROFIT,DATA(30,70),SAVE(100,4),ATTN
      COMMON MINPRF,MAXPRF,MAXPTR,YNDEF(4),REP(3),NUMNUM,NUMHLP
      COMMON NUMLIT,NUMGET,NUMCMD,NUMDEF,NUMERR,MAXCHR,NUMCHR,RZ
      COMMON NUMNDA,NUMNDN
C
      GOTO(1,2,3,4,5,6,7,5,9,99,5,12,5,14,15,16,17,18,19,5,21,
     * 5,23,5,25,26,27,28,29,30,31,32,33,5,34,5,36,5,38,5,40,
     * 5,42,5,44,5,46,48,49), MSG
    1 WRITE(6,101)
  101 FORMAT(10(/),
     * 'You are the General Manager for a small company ',
     * 'called XYZ (name '/
     * 'disguised), which manufactures and sells one product, ',
     * 'Widgets (again '/
     * 'disguised). In your continuing efforts to meet ',
     * 'company objectives '/
     * ' - i.e. to maximize profit (what else!!!) - ',
     * 'you recently hired an '/
     * 'M.B.A. student, John Doe, to undertake some ',
     * 'quantitative analysis.'//
     * 'John was instructed to develop a model and ',
     * 'computer program to help '/
     * 'find the optimal Retail Price and Production ',
     * 'Quantity for Widgets.'/
     * 'After weeks of diligent work he has produced a ',
     * 'very "sophisticated"'/
     * 'WATFIV program to do the job.'/)
      WRITE(6,201)
  201 FORMAT('It is Monday morning, and John is waiting ',
     * 'for you when you arrive'/
     * 'at the office. He proudly presents his work to you. ',
     * 'Unfortunately,'/
     * 'being from a famous Eastern Business School, ',
     * 'he never thought to'/
     * 'use the computer to actually determine ',
     * 'the optimum automatically;'/
     * 'instead, he designed a program with which ',
     * 'you could seek the opti-'/
     * 'mum yourself (by spending precious time at ',
     * 'a computer terminal,'/
     * 'simulating the results of different ',
     * 'Price/Quantity combinations).'/)
      WRITE(6,301)
  301 FORMAT('You refrain from strangling John, ',
     * 'and calmly thank him for his'/
     * 'efforts (while making a mental memo ',
     * 'to hire only U.B.C. graduates'/
     * 'in the future). You then proceed to ',
```

```
     * 'the Computing Centre to try'/
     * 'out the new program.'//
     * ' As you arrive at the terminal room, ',
     * 'you recall your marketing manager''s'/
     * ' report indicating that your firm''s demand',
     * ' function is rather unusual.'/
     * ' You make a mental note nct to let your ',
     * 'intuition lead you astray,'/
     * ' and then start running the program...'///
     * '&Press RETURN to continue.')
      RETURN
    2 WRITE(6,102)
  102 FORMAT(25(/),'1*** THE PROGRAM ***'//
     * ' The simulaticn is directed by you, the user.'/
     * ' When the word "COMMAND :" appears, either enter a command'/
     * ' or just press RETURN to get a list of available commands.'/
     * ' Remember: All commands may be typed in full OR abbreviated '/
     * ' as you wish.'//
     * ' Some helpful hints:'/
     * '      1. The possible price range is 1-30.'/
     * '      2. The possible quantity range is 1-70.'/
     * '      3. There is one and only one maximum point.'/
     * '      4. The game will automatically stop after 25 minutes.'/
     * '      5. The game will alsc stop when you find the optimum.'/
     * '      6. The optimum values are different for everyone!'/
     * '      7. After a few periods, be sure to try all reports'/
     * '         in order to learn what they are...'///)
      RETURN
    3 WRITE(6,103)
  103 FORMAT(25(/),'1*** THE PROGRAM ***'//
     * ' The program will guide you through the simulation, ',
     * 'step by step.'/
     * ' Simply answer all questions as directed.'//
     * ' Some helpful hints:'/
     * '      1. The possible price range is 1-30.'/
     * '      2. The possible quantity range is 1-70.'/
     * '      3. There is one and only one maximum point.'/
     * '      4. The game will autcmatically stop after 25 minutes.'/
     * '      5. The game will also stop when you find the optimum.'/
     * '      6. The optimum values are different for everyone!'/
     * '      7. After a few periods, be sure to try all reports'/
     * '         in order to learn what they are...'///)
      RETURN
    4 WRITE(6,104)
  104 FORMAT(15(1X,78('$')/),1X,15('$'),48X,15('$')/1X,15('$'),2X,
     * 'CONGRATULATIONS! YOU HAVE FOUND THE MAXIMUM!',2X,
     * 15('$')/1X,15('$'),48X,15('$')/1X,15('$'),2X,
     * 'PLEASE TELL THE SUPERVISOR THAT YOU ARE DONE',2X,
     * 15('$')/1X,15('$'),48X,15('$')/15(1X,78('$')/))
      RETURN
    5 WRITE(6,105)
  105 FORMAT('1*** ILLEGAL INPUT ***    Try again...'/)
      RETURN
    6 WRITE(6,106)
  106 FORMAT(/'&ENTER LOWEST PRICE TO BE DISPLAYED (1-26)')
      RETURN
    7 WRITE(6,107)
```

```
107 FORMAT('0No reports until you have begun playing!!'/)
    RETURN
  9 WRITE(6,109) PERIOD,PRICE,QTY,PROFIT
109 FORMAT(//' Period ',I3,' has been simulated...'//
   * ' With PRICE=',I3,'  and QUANTITY=',I3,'  your profit was  $'
   * ,I2///)
    RETURN
 12 WRITE(6,112) PRICE
112 FORMAT(/'&Enter desired price level (1-30) [',I3,']')
    RETURN
 14 WRITE(6,114) QTY
114 FORMAT(/'&Enter desired quantity produced (1-70) [',I3,']')
    RETURN
 15 WRITE(6,115) (YNDEF(I),I=1,3)
115 FORMAT(/'&Want to see History Report (YES or NO) [',3A1,']?')
    RETURN
 16 WRITE(6,116) (YNDEF(I),I=1,3)
116 FORMAT(/'&Want to see Ordered Report (YES or NO) [',3A1,']?')
    RETURN
 17 WRITE(6,117) (YNDEF(I),I=1,3)
117 FORMAT(/'&Want to see Summary Graph (YES or NO) [',3A1,']?')
    RETURN
 18 WRITE(6,118)
118 FORMAT(//' ***** Only available commands are:'//
   * 'PRICE         Set retail price for this period'/
   * 'QUANTITY      Set production quantity for this period'/
   * 'SIMULATE      Simulate this period''s results'/
   * 'HISTORY       Provide History Report'/
   * 'ORDERING      Provide Ordered History Report'/
   * 'GRAPH         Provide Summary Graph'//)
    RETURN
 19 WRITE(6,119)
119 FORMAT(/'&COMMAND')
    RETURN
 21 WRITE(6,121)
121 FORMAT(' If 100 other people were playing this game right now,'
   * /'&  how many would be closer to the optimum than you (0-100)?')
    RETURN
 23 WRITE(6,123)
123 FORMAT(/' How would you rate the "usability" of this program;'
   * /'&  from 1 to 9, where 1=frustrating, 9=convenient (1-9)?')
    RETURN
 25 WRITE(6,125)
125 FORMAT(/' How would you describe your present attitude'
   * /'&  toward this game; 1=bored, 9=enjoying it (1-9)?')
    RETURN
 26 WRITE(6,126)
126 FORMAT(/1X,65(':')//
   * ' Please CAREFULLY answer the following three questions:'/)
    RETURN
 27 WRITE(6,127)
127 FORMAT(/1X,65(':')///)
    RETURN
 28 WRITE(6,128)
128 FORMAT(32(/),' History Report for most recent 25 periods.'//
   * '    PERIOD   PRICE     QTY   PROFIT'/)
    RETURN
```

```
 29 WRITE(6,129)
129 FORMAT(32(/),' History Report - ordered by Profit.'//
   * '   PERIOD   PRICE     QTY  PROFIT'/)
    RETURN
 30 WRITE(6,130)
130 FORMAT(///23X,'Graph of PROFIT/10  vs. PRICE,QTY')
    RETURN
 31 WRITE(6,131)
131 FORMAT(//' ***** Only available commands are:'//
   * 'PRICE          Set retail price for this period'/
   * 'QTY            Set production quantity for this period'/
   * 'SIM            Simulate this period''s results'/
   * 'HIST           Provide History Report'/
   * 'ORD            Provide Ordered History Report'/
   * 'GRAPH          Provide Summary Graph'//)
    RETURN
 32 WRITE(6,132)
132 FORMAT(15(1X,78('$')/),1X,15('$'),48X,15('$')/1X,15('$'),2X,
   * 'YOU HAVE EXITTED WITH AN ATTENTION INTERUPT.',2X,
   * 15('$')/1X,15('$'),48X,15('$')/1X,15('$'),2X,
   * 'PLEASE TELL THE SUPERVISOR THAT YOU ARE DONE',2X,
   * 15('$')/1X,15('$'),48X,15('$')/15(1X,78('$')/))
    RETURN
 33 WRITE(6,133)
133 FORMAT(15(1X,78('$')/),1X,15('$'),48X,15('$')/1X,15('$'),2X,
   * ' SORRY, YOU HAVE EXCEEDED THE MAXIMUM TIME. ',2X,
   * 15('$')/1X,15('$'),48X,15('$')/1X,15('$'),2X,
   * 'PLEASE TELL THE SUPERVISOR THAT YOU ARE DONE',2X,
   * 15('$')/1X,15('$'),48X,15('$')/15(1X,78('$')/))
    RETURN
 34 WRITE(6,134)
134 FORMAT(/' If 100 other people had played this game, how many'
   * /'&  would have found the optimum in fewer periods (0-100)?')
    RETURN
 36 WRITE(6,136)
136 FORMAT(/' If 100 other people had played this game, how many'
   * /'&  would have found the optimum in less time (0-100)?')
    RETURN
 38 WRITE(6,138)
138 FORMAT(//' *** For the next 3 questions, "9" is best ***'/
   * /'&How useful was the History Report, from 1 to 9 (1-9)?')
    RETURN
 40 WRITE(6,140)
140 FORMAT(/'&How useful was the Ordered History Report (1-9)?')
    RETURN
 42 WRITE(6,142)
142 FORMAT(/'&How useful was the Graph Report (1-9)?')
    RETURN
 44 WRITE(6,144)
144 FORMAT(/' In your search for the optimum, about how many'
   * /'   periods did it take you to zoom in on the general'
   * /'&  vicinity of the optimum PRICE,QTY pair (1-50)?')
    RETURN
 46 WRITE(6,146)
146 FORMAT(/' Would you describe your search for the optimum as'
   * /'   reasonably direct&structured (enter "1") or rather'
   * /'&  random&haphazard (enter "2") (1-2)?')
```

```
      RETURN
   48 WRITE(6,148)
  148 FORMAT(///////////////////' *** THANK YOU FOR PARTICIPATING ***'/
     * ///' Please refrain from discussing the game with others until'
     * //' after March 31st.'///)
      RETURN
   49 WRITE(6,149)
  149 FORMAT(//'1POST-GAME QUESTIONNAIRE'/' ========= ============='//)
   99 RETURN
      END
```

```
      SUBROUTINE READPF(NAME,MX,MY,RX)
C
C     READS IN PROFIT FUNCTION
C     (AFTER RANDOMLY SETTING LOCATION OF OPTIMAL POINT)
C
      IMPLICIT INTEGER(A-Z)
      INTEGER*2 NAME(6)
      LOGICAL*1 YNDEF
      REAL RX
      INTEGER*2 DATA,SAVE
      LOGICAL ATTN
      REAL RZ
      COMMON PERIOD,PRICE,QTY,PROFIT,DATA(30,70),SAVE(100,4),ATTN
      COMMON MINPRF,MAXPRF,MAXPTR,YNDEF(4),REP(3),NUMNUM,NUMHLP
      COMMON NUMLIT,NUMGET,NUMCMD,NUMDEF,NUMERR,MAXCHR,NUMCHR,RZ
      COMMON NUMNDA,NUMNDN
C
      IF(MX.NE.0) GOTO 60
      CALL TIME(2,0,MX)
      MX=-MX
      MX=IRAND(MX)
   51 MX=IRAND(0)
      MX=IRAND(10)
      IF(MX.GT.5) MX=MX+7
      MY=IRAND(20)
      IF(MY.GT.10) MY=MY+7
      MZ=IRAND(375)
      RX=(MZ+870.0)/1000.0
      WRITE(7,55) (NAME(I),I=1,6),MX,MY,RX
   55 FORMAT(' ONAME=',6A2,'  MX=',I3,'  MY=',I3,'  RZ=',F5.3)
   60 DO 61 I=1,MY
         READ(8,65)
   61    CONTINUE
      DO 66 J=1,70
         READ(8,65) (DATA(I,1),I=1,MX),(DATA(I,J),I=1,30)
   65    FORMAT(47I3)
   66    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SGRAPH
C
C     OUTPUTS A GRAPH OF PROFIT/10  VS.  PRICE,QTY
C
      IMPLICIT INTEGER(A-Z)
      INTEGER PRVPER/1/
      INTEGER*2 SCR(30,70)/2100*1/
      LOGICAL*1 SHADE(12)/' ','0','1','2','3','4','5','6','7','8',
     * '9','.'/,YNDEF,FIRST/.TRUE./
      LOGICAL*1 PLABEL(30)/11*'X',' ','E','C','I','R','P',' ',12*'X'/
      INTEGER*2 DATA,SAVE
      LOGICAL ATTN
      REAL RZ
      COMMON PERIOD,PRICE,QTY,PROFIT,DATA(30,70),SAVE(100,4),ATTN
      COMMON MINPRF,MAXPRF,MAXPTR,YNDEF(4),REP(3),NUMNUM,NUMHLP
      COMMON NUMLIT,NUMGET,NUMCMD,NUMDEF,NUMERR,MAXCHR,NUMCHR,RZ
      COMMON NUMNDA,NUMNDN
C
C     INITIALIZATION
C
      REP(3)=REP(3)+1
      IF(PERIOD.GT.0)  GOTO 1
      CALL OUTMES(7)
      GOTO 50
    1 CALL OUTMES(30)
      IF (.NOT.FIRST) GOTO 6
      DO 3 I=1,30
          DO 2 J=5,70,5
              SCR(I,J)=12
    2         CONTINUE
    3     CONTINUE
      DO 5 I=5,30,5
          DO 4 J=1,70
              SCR(I,J)=12
    4         CONTINUE
    5     CONTINUE
      FIRST=.FALSE.
C
C     SET UP SCREEN MATRIX
C
    6 DO 10 I=PRVPER,PERIOD
          SCR(SAVE(I,1),SAVE(I,2))=SAVE(I,3)/10+2
   10     CONTINUE
C
C     NOW DRAW THE GRAPH
C
      DO 40 J=1,30
          I=31-J
          WRITE(6,30) I,PLABEL(I),(SHADE(SCR(I,L)),L=1,70)
   30     FORMAT(1X,I3,1X,71A1)
   40     CONTINUE
      WRITE(6,45)
   45 FORMAT(5X,31('X'),' QUANTITY ',30('X')//6X,'123456789',
     * 10('1'),10('2'),10('3'),10('4'),10('5'),10('6'),'7'/
     * 15X,6('0123456789'),'0')
   50 RETURN
      END
```

```
      SUBROUTINE SIMUL(RECOVR,MTIM)
C
C     SIMULATE ANOTHER PERIOD OF PLAY
C
      IMPLICIT INTEGER(A-Z)
      LOGICAL RECOVR
      INTEGER TIMNEW,TIMOLD/0/,TOTTIM/0/
      LOGICAL*1 YNDEF
      INTEGER*2 DATA,SAVE
      LOGICAL ATTN
      REAL RZ
      COMMON PERIOD,PRICE,QTY,PROFIT,DATA(30,70),SAVE(100,4),ATTN
      COMMON MINPRF,MAXPRF,MAXPTR,YNDEF(4),REP(3),NUMNUM,NUMHLP
      COMMON NUMLIT,NUMGET,NUMCMD,NUMDEF,NUMERR,MAXCHR,NUMCHR,RZ
      COMMON NUMNDA,NUMNDN
C
C     CLEAR TYPEAHEAD AND RUN SIMPLE SIMULATION
C
      CALL CLRSTR
      PERIOD=PERIOD+1
      PROFIT=RZ*DATA(PRICE,QTY)
      IF(MINPRF.GT.PROFIT) MINPRF=PROFIT
      IF(MAXPRF.LT.PROFIT) MAXPRF=PROFIT
      SAVE(PERIOD,1)=PRICE
      SAVE(PERIOD,2)=QTY
      SAVE(PERIOD,3)=PROFIT
      IF(RECOVR) TOTTIM=TOTTIM+MTIM
      IF(RECOVR) GOTO 15
      CALL OUTMES(9)
C
C     GET CONNECT TIME OF USER
C
      CALL TIME(2,0,TIMNEW)
      TIMNEW=TIMNEW/100
      TIM=TIMNEW-TIMOLD
      TOTTIM=TOTTIM+TIM
      TIMOLD=TIMNEW
      WRITE(7,10) PERIOD,PRICE,QTY,PROFIT,NUMLIT,NUMGET,NUMCMD,NUMDEF,
     * NUMERR,MAXCHR,NUMCHR,NUMNUM,NUMHLP,NUMNDA,NUMNDN,
     * (REP(I),I=1,3),TIM
   10 FORMAT(' 1',4I3,12I2,2I1,I5)
C
C     GET USER ATTITUDES (ONLY IN EACH 10TH PERIOD)
C
      IF(MOD((PERIOD-5),10).NE.0) GOTO 15
      I1=-1
      I2=-1
      I3=-1
      CALL OUTMES(26)
      CALL GETNUM(.TRUE.,I1,0,100,20,21)
      CALL GETNUM(.TRUE.,I2,1,9,22,23)
      CALL GETNUM(.TRUE.,I3,1,9,24,25)
      WRITE(7,12) I1,I2,I3
   12 FORMAT(' 2',3I3)
      CALL OUTMES(27)
      CALL TIME(2,0,TIMOLD)
      TIMOLD=TIMOLD/100
```

```
C
C      PUT NEW RECORD INTO SORTED CHAIN
C
   15 IF(PERIOD.EQ.1) GOTO 50
      IF(PROFIT.LT.SAVE(MAXPTR,3)) GOTO 20
      SAVE(PERIOD,4)=MAXPTR
      MAXPTR=PERIOD
      GOTO 50
   20 OLDPTR=MAXPTR
      PTR=SAVE(OLDPTR,4)
      DO 30 I=1,499
          IF(PTR.EQ.0) GOTO 40
          IF(PROFIT.GE.SAVE(PTR,3)) GOTO 40
          OLDPTR=PTR
          PTR=SAVE(OLDPTR,4)
   30     CONTINUE
   40 SAVE(PERIOD,4)=PTR
      SAVE(OLDPTR,4)=PERIOD
C
C      CHECK FOR END-OF-GAME
C
   50 IF(DATA(PRICE,QTY).NE.80 .AND. .NOT.ATTN .AND. TOTTIM.LT.15000)
     *    GOTO 60
      IF(DATA(PRICE,QTY).EQ.80) GOTO 52
      IF(ATTN) GOTO 54
      IF(TOTTIM.GE.15000) GOTO 56
   52 CALL OUTMES(4)
      IWAY=1
      GOTO 58
   54 CALL OUTMES(32)
      IWAY=2
      GOTO 58
   56 CALL OUTMES(33)
      IWAY=3
   58 CALL ATNTRP(ATTN)
      CALL BELLWT(1)
      CALL ZEND(IWAY)
C
C      RESET ALL COUNTERS
C
   60 NUMLIT=0
      NUMGET=0
      NUMCMD=0
      NUMDEF=0
      NUMERR=0
      MAXCHR=0
      NUMCHR=0
      NUMNUM=0
      NUMHLP=0
      NUMNDA=0
      NUMNDN=0
      REP(1)=0
      REP(2)=0
      REP(3)=0
   70 RETURN
      END
```

```
      SUBROUTINE SORTH
C
C     OUTPUT SORTED RESULTS
C
      IMPLICIT INTEGER(A-Z)
      LOGICAL*1 YNDEF
      INTEGER*2 DATA,SAVE
      LOGICAL ATTN
      REAL RZ
      COMMON PERIOD,PRICE,QTY,PROFIT,DATA(30,70),SAVE(100,4),ATTN
      COMMON MINPRF,MAXPRF,MAXPTR,YNDEF(4),REP(3),NUMNUM,NUMHLP
      COMMON NUMLIT,NUMGET,NUMCMD,NUMDEF,NUMERR,MAXCHR,NUMCHR,RZ
      COMMON NUMNDA,NUMNDN
C
      REP(2)=REP(2)+1
      IF(PERIOD.GT.0) GOTO 5
      CALL OUTMES(7)
      GOTO 30
    5 K=25
      IF(PERIOD.LT.K) K=PERIOD
      PTR=MAXPTR
      CALL OUTMES(29)
      DO 20 I=1,K
          WRITE(6,10) PTR,SAVE(PTR,1),SAVE(PTR,2),SAVE(PTR,3)
   10     FORMAT(1X,4I8)
          PTR=SAVE(PTR,4)
   20     CONTINUE
   30 RETURN
      END
```

```
      SUBROUTINE ZEND(IWAY)
C
C     END-OF-GAME CLEANUP
C
      IMPLICIT INTEGER(A-Z)
      INTEGER Q(9)/9*-1/
      LOGICAL*1 YNDEF
      INTEGER*2 DATA,SAVE
      LOGICAL ATTN
      REAL RZ
      COMMON PERIOD,PRICE,QTY,PROFIT,DATA(30,70),SAVE(100,4),ATTN
      COMMON MINPRF,MAXPRF,MAXPTR,YNDEF(4),REP(3),NUMNUM,NUMHLP
      COMMON NUMLIT,NUMGET,NUMCMD,NUMDEF,NUMERR,MAXCHR,NUMCHR,RZ
      COMMON NUMNDA,NUMNDN
C
      CALL OUTMES(49)
      CALL GETNUM(.TRUE.,Q(1),0,100,34,35)
      CALL GETNUM(.TRUE.,Q(2),0,100,36,37)
      CALL GETNUM(.TRUE.,Q(3),1,9,22,23)
      CALL GETNUM(.TRUE.,Q(4),1,9,24,25)
      CALL GETNUM(.TRUE.,Q(5),1,9,38,39)
      CALL GETNUM(.TRUE.,Q(6),1,9,40,41)
      CALL GETNUM(.TRUE.,Q(7),1,9,42,43)
      CALL GETNUM(.TRUE.,Q(8),1,50,44,45)
      CALL GETNUM(.TRUE.,Q(9),1,2,46,47)
      CALL OUTMES(48)
      WRITE(7,10) IWAY,(Q(I),I=1,9)
   10 FORMAT(' 3',4I3,12I2,2I1,I5)
      CALL RTWAIT(1500)
      CALL CMD('COPY PSM#1 TO IZAK:REPS(LAST+1) ',32)
      CALL CMD('SIG ',4)
      STOP
      END
```

Appendix B

GAME INSTRUCTIONS

Listings of the pre-game instructions appear on the next few pages; instructions for the structured game version are on the next page, and instructions for the unstructured game are on the two pages following that. As can be seen, the participant is only given directions for using the computer terminal and special program features; the exact nature of the game is described when the game is actually played (see Appendix C).

INSTRUCTIONS

You will soon be playing a simple computer game (a "simulation"). The nature of the game will be described in detail when you begin playing. In the meantime, please read (and understand!) the following instructions - they are short, so please read them at least a few times:

1) To enter input into the computer, simply type on the computer terminal keyboard as if it were a normal typewriter. After you have enterred a line, press the RETURN key to terminate the input.

2) If you make a typing mistake in the current line, just press the DEL LINE key (near the top right) and then retype the line.

3) All questions asked by the game are of the same format; the following example illustrates it:

```
Want to see the History Report (YES or NO) [YES]? :
```

As can be seen, first the actual question is displayed, followed by the range of possible answers in parentheses, followed - in brackets - by the answer which the computer will assume you want if you simply press the RETURN key. To answer NO to the above question, you could type NO or N - and then press RETURN. To answer YES, you could type YES, Y, or nothing at all - and then press RETURN.

4) Finally, there is one report which must be explained. It is a 3-dimensional graph, and is best explained with an example:

```
PRICE  QTY  PROFIT                     5 |  2 0
-----  ---  ------                     4 | 3
  3     4     20                 PRICE 3 |    2
  5     2     23                       2 |
  1     2     17       ===>            1 |  1
  4     1     33                         |
  5     4     08                          12345
                                           QTY
```

As can be seen, PRICE is the vertical axis, QTY is the horizontal axis, and the PROFIT is represented by a single digit (PROFIT/10 - no rounding!) at the intersection of the associated PRICE,QTY pair.

INSTRUCTIONS

You will soon be playing a simple computer game (a "simulation"). The nature of the game will be described in detail when you begin playing. In the meantime, please read (and understand!) the following instructions - they are short, so please read them at least a few times:

1) To enter input into the computer, simply type on the computer terminal keyboard as if it were a normal typewriter. After you have enterred a line, press the RETURN key to terminate the input.

2) If you make a typing mistake in the current line, just press the DEL LINE key (near the top right) and then retype the line.

3) You will have to take the initiative in this game; that is, you will have to instruct the computer what to do next. To do this, you must enter commands via the keyboard (the commands will be described when you play). When you enter commands, you can type the entire command, or any abbreviation of it. Thus, to enter the command SIMULAT you could type SIMULATE, SIMUL, SIM, S, etc. - and then press RETURN.

4) Some commands will cause a question to be asked by the computer. All questions asked will be of the same format; the following example illustrates it:

```
    Enter price to be charged next period (1-30) [10] :
```

As can be seen, first the actual question is displayed, followed by the range of possible answers in parentheses, followed - in brackets - by the answer which the computer will assume you want if you simply press the RETURN key. To answer 20 to the above question, you could type 20 - and then press RETURN. To answer 10, you could type 10, or nothing at all - and then press RETURN.

5) You may also combine commands on one line (separated by spaces!) if you wish. For example, if you knew that the following sequence of events would occur (note that all lines end with a RETURN):

```
    Command : PRICE
      Enter price to be charged next period (1-30) [10] : 20
    Command : SIMUL
```

you could have just typed :

```
    Command : PRICE 20
    Command : SIMUL
```

or even:

```
    Command : PRICE 20 SIMUL
```

6) Finally, there is one report which must be explained. It is a 3-dimensional graph, and is best explained with an example:

PRICE	QTY	PROFIT
3	4	20
5	2	23
1	2	17
4	1	33
5	4	08

===>

```
        5 |  2 0
        4 | 3
  PRICE 3 |    2
        2 |
        1 |  1
          |
           12345
            QTY
```

As can be seen, PRICE is the vertical axis, QTY is the horizontal axis, and the PROFIT is represented by a single digit (PROFIT/10 - no rounding!) at the intersection of the associated PRICE,QTY pair.

Appendix C

SAMPLE INTERACTION

The next pages provide examples of two sessions of the computer game (a structured version interaction appears on the first 6 pages, while an unstructured version interaction appears on the 4 pages following those). The opening instructions, several periods of simulation, an attitude questionnaire, and all three reports are presented. (Note that the graphs are much more readable on the computer terminal where the dots are much fainter).

You are the General Manager for a small company called XYZ (name disguised), which manufactures and sells one product, Widgets (again disguised). In your continuing efforts to meet company objectives - i.e. to maximize profit (what else!!!) - you recently hired an M.B.A. student, John Doe, to undertake some quantitative analysis.

John was instructed to develop a model and computer program to help find the optimal Retail Price and Production Quantity for Widgets. After weeks of diligent work he has produced a very "sophisticated" WATFIV program to do the job.

It is Monday morning, and John is waiting for you when you arrive at the office. He proudly presents his work to you. Unfortunately, being from a famous Eastern Business School, he never thought to use the computer to actually determine the optimum automatically; instead, he designed a program with which you could seek the optimum yourself (by spending precious time at a computer terminal, simulating the results of different Price/Quantity combinations).

You refrain from strangling John, and calmly thank him for his efforts (while making a mental memo to hire only U.B.C. graduates in the future). You then proceed to the Computing Centre to try out the new program.

As you arrive at the terminal room, you recall your marketing manager report indicating that your firm's demand function is rather unusual. You make a mental note not to let your intuition lead you astray, and then start running the program...

*** THE PROGRAM ***

The program will guide you through the simulation, step by step. Simply answer all questions as directed.

Some helpful hints:

1. The possible price range is 1-30.
2. The possible quantity range is 1-70.
3. There is one and only one maximum point.
4. The game will automatically stop after 25 minutes.
5. The game will also stop when you find the optimum.
6. The optimum values are different for everyone!
7. After a few periods, be sure to try all reports in order to learn what they are...

```
Enter desired price level (1-30) [ 10]  : 15
Enter desired quantity produced (1-70) [ 25]  :

Period   1 has been simulated...
With PRICE= 15  and QUANTITY= 25  your profit was  $27


Want to see History Report (YES or NO) [NO ]?  : NO
Want to see Ordered Report (YES or NO) [NO ]?  : N
Want to see Summary Graph (YES or NO) [NO ]?  :
Enter desired price level (1-30) [ 15]  :
Enter desired quantity produced (1-70) [ 25]  : 35

Period   2 has been simulated...
With PRICE= 15  and QUANTITY= 35  your profit was  $64


Want to see History Report (YES or NO) [NO ]?  :
Want to see Ordered Report (YES or NO) [NO ]?  :
Want to see Summary Graph (YES or NO) [NO ]?  :
Enter desired price level (1-30) [ 15]  :
Enter desired quantity produced (1-70) [ 35]  : 45

Period   3 has been simulated...
With PRICE= 15  and QUANTITY= 45  your profit was  $77


Want to see History Report (YES or NO) [NO ]?  :
Want to see Ordered Report (YES or NO) [NO ]?  :
Want to see Summary Graph (YES or NO) [NO ]?  :
Enter desired price level (1-30) [ 15]  : 10
Enter desired quantity produced (1-70) [ 45]  : 35
```

```
Period   4 has been simulated...

With PRICE= 10  and QUANTITY= 35  your profit was  $43


Want to see History Report (YES or NO) [NO ]?  :

Want to see Ordered Report (YES or NO) [NO ]?  :

Want to see Summary Graph (YES or NO) [NO ]?  :

Enter desired price level (1-30) [ 10]  : 20

Enter desired quantity produced (1-70) [ 35]  :

Period   5 has been simulated...

With PRICE= 20  and QUANTITY= 35  your profit was  $43


::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::

Please CAREFULLY answer the following three questions:

If 100 other people were playing this game right now,
  how many would be closer to the optimum than you (0-100)?  : 25

How would you rate the "usability" of this program;
  from 1 to 9, where 1=frustrating, 9=convenient (1-9)?  : 7

How would you describe your present attitude
  toward this game; 1=bored, 9=enjoying it (1-9)?  : 9

::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::


Want to see History Report (YES or NO) [NO ]?  : YES


History Report for most recent 25 periods.

  PERIOD   PRICE     QTY  PROFIT

       1       15      25       27
       2       15      35       64
       3       15      45       77
       4       10      35       43
       5       20      35       43
```

Want to see Ordered Report (YES or NO) [NO]? : Y

History Report - ordered by Profit.

PERIOD	PRICE	QTY	PROFIT
3	15	45	77
2	15	35	64
5	20	35	43
4	10	35	43
1	15	25	27

Want to see Summary Graph (YES or NO) [NO]? : Y

```
                         Graph of PROFIT/10  vs.  PRICE,QTY
30 X.................................................................
29 X    .    .    .    .    .    .    .    .    .    .    .    .    .
28 X    .    .    .    .    .    .    .    .    .    .    .    .    .
27 X    .    .    .    .    .    .    .    .    .    .    .    .    .
26 X    .    .    .    .    .    .    .    .    .    .    .    .    .
25 X.................................................................
24 X    .    .    .    .    .    .    .    .    .    .    .    .    .
23 X    .    .    .    .    .    .    .    .    .    .    .    .    .
22 X    .    .    .    .    .    .    .    .    .    .    .    .    .
21 X    .    .    .    .    .    .    .    .    .    .    .    .    .
20 X..................................4..............................
19 X    .    .    .    .    .    .    .    .    .    .    .    .    .
18      .    .    .    .    .    .    .    .    .    .    .    .    .
17 P    .    .    .    .    .    .    .    .    .    .    .    .    .
16 R    .    .    .    .    .    .    .    .    .    .    .    .    .
15 I........................2.........6.........7....................
14 C    .    .    .    .    .    .    .    .    .    .    .    .    .
13 E    .    .    .    .    .    .    .    .    .    .    .    .    .
12      .    .    .    .    .    .    .    .    .    .    .    .    .
11 X    .    .    .    .    .    .    .    .    .    .    .    .    .
10 X..................................4..............................
 9 X    .    .    .    .    .    .    .    .    .    .    .    .    .
 8 X    .    .    .    .    .    .    .    .    .    .    .    .    .
 7 X    .    .    .    .    .    .    .    .    .    .    .    .    .
 6 X    .    .    .    .    .    .    .    .    .    .    .    .    .
 5 X.................................................................
 4 X    .    .    .    .    .    .    .    .    .    .    .    .    .
 3 X    .    .    .    .    .    .    .    .    .    .    .    .    .
 2 X    .    .    .    .    .    .    .    .    .    .    .    .    .
 1 X    .    .    .    .    .    .    .    .    .    .    .    .    .
    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX QUANTITY XXXXXXXXXXXXXXXXXXXXXXXXX
    12345678911111111112222222222333333333344444444445555555555666666
             01234567890123456789012345678901234567890123456789012345
```

Enter desired price level (1-30) [20] :

Enter desired quantity produced (1-70) [35] : 45

```
Period   6 has been simulated...

With PRICE= 20  and QUANTITY= 45  your profit was  $39


Want to see History Report (YES or NO) [NO ]?  :

Want to see Ordered Report (YES or NO) [NO ]?  :

Want to see Summary Graph (YES or NO) [NO ]?  :

Enter desired price level (1-30) [ 20]  : 10

Enter desired quantity produced (1-70) [ 45]  : 45

Period   7 has been simulated...

With PRICE= 10  and QUANTITY= 45  your profit was  $55


Want to see History Report (YES or NO) [NO ]?  :

Want to see Ordered Report (YES or NO) [NO ]?  :

Want to see Summary Graph (YES or NO) [NO ]?  :

Enter desired price level (1-30) [ 10]  : 15

Enter desired quantity produced (1-70) [ 45]  : 55

Period   8 has been simulated...

With PRICE= 15  and QUANTITY= 55  your profit was  $30


Want to see History Report (YES or NO) [NO ]?  :

Want to see Ordered Report (YES or NO) [NO ]?  :

Want to see Summary Graph (YES or NO) [NO ]?  : Y
```

```
                    Graph of PROFIT/10  vs.  PRICE,QTY
30 X.................................................................
29 X    .    .    .    .    .    .    .    .    .    .    .    .    .
28 X    .    .    .    .    .    .    .    .    .    .    .    .    .
27 X    .    .    .    .    .    .    .    .    .    .    .    .    .
26 X    .    .    .    .    .    .    .    .    .    .    .    .    .
25 X.................................................................
24 X    .    .    .    .    .    .    .    .    .    .    .    .    .
23 X    .    .    .    .    .    .    .    .    .    .    .    .    .
22 X    .    .    .    .    .    .    .    .    .    .    .    .    .
21 X    .    .    .    .    .    .    .    .    .    .    .    .    .
20 X..................................4.........3....................
19 X    .    .    .    .    .    .    .    .    .    .    .    .    .
18      .    .    .    .    .    .    .    .    .    .    .    .    .
17 P    .    .    .    .    .    .    .    .    .    .    .    .    .
16 R    .    .    .    .    .    .    .    .    .    .    .    .    .
15 I........................2.........6.........7.........3..........
14 C    .    .    .    .    .    .    .    .    .    .    .    .    .
13 E    .    .    .    .    .    .    .    .    .    .    .    .    .
12      .    .    .    .    .    .    .    .    .    .    .    .    .
11 X    .    .    .    .    .    .    .    .    .    .    .    .    .
10 X..................................4.........5....................
 9 X    .    .    .    .    .    .    .    .    .    .    .    .    .
 8 X    .    .    .    .    .    .    .    .    .    .    .    .    .
 7 X    .    .    .    .    .    .    .    .    .    .    .    .    .
 6 X    .    .    .    .    .    .    .    .    .    .    .    .    .
 5 X.................................................................
 4 X    .    .    .    .    .    .    .    .    .    .    .    .    .
 3 X    .    .    .    .    .    .    .    .    .    .    .    .    .
 2 X    .    .    .    .    .    .    .    .    .    .    .    .    .
 1 X    .    .    .    .    .    .    .    .    .    .    .    .    .
   XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX QUANTITY XXXXXXXXXXXXXXXXXXXXXXXXX

    12345678911111111112222222222333333333344444444445555555555666666
             01234567890123456789012345678901234567890123456789012345
```

Enter desired price level (1-30) [15] :

Enter desired quantity produced (1-70) [55] : 40

Period   9 has been simulated...

With PRICE= 15 and QUANTITY= 40 your profit was $70

You are the General Manager for a small company called XYZ (name disguised), which manufactures and sells one product, Widgets (again disguised). In your continuing efforts to meet company objectives - i.e. to maximize profit (what else!!!) - you recently hired an M.B.A. student, John Doe, to undertake some quantitative analysis.

John was instructed to develop a model and computer program to help find the optimal Retail Price and Production Quantity for Widgets. After weeks of diligent work he has produced a very "sophisticated" WATFIV program to do the job.

It is Monday morning, and John is waiting for you when you arrive at the office. He proudly presents his work to you. Unfortunately, being from a famous Eastern Business School, he never thought to use the computer to actually determine the optimum automatically; instead, he designed a program with which you could seek the optimum yourself (by spending precious time at a computer terminal, simulating the results of different Price/Quantity combinations).

You refrain from strangling John, and calmly thank him for his efforts (while making a mental memo to hire only U.B.C. graduates in the future). You then proceed to the Computing Centre to try out the new program.

As you arrive at the terminal room, you recall your marketing manager report indicating that your firm's demand function is rather unusual. You make a mental note not to let your intuition lead you astray, and then start running the program...

*** THE PROGRAM ***

The simulation is directed by you, the user.
When the word "COMMAND :" appears, either enter a command or just press RETURN to get a list of available commands.
Remember: All commands may be typed in full OR abbreviated as you wish.

Some helpful hints:

1. The possible price range is 1-30.
2. The possible quantity range is 1-70.
3. There is one and only one maximum point.
4. The game will automatically stop after 25 minutes.
5. The game will also stop when you find the optimum.
6. The optimum values are different for everyone!
7. After a few periods, be sure to try all reports in order to learn what they are...

```
***** Only available commands are:

PRICE         Set retail price for this period
QUANTITY      Set production quantity for this period
SIMULATE      Simulate this period's results
HISTORY       Provide History Report
ORDERING      Provide Ordered History Report
GRAPH         Provide Summary Graph


COMMAND  : PRICE

Enter desired price level (1-30) [ 10]  : 15

COMMAND  : QUANTITY

Enter desired quantity produced (1-70) [ 25]  :

COMMAND  : SIMULATE


Period   1 has been simulated...

With PRICE= 15  and QUANTITY= 25  your profit was  $35



COMMAND  : PRICE 15 QUANTITY 35

COMMAND  : SIM


Period   2 has been simulated...

With PRICE= 15  and QUANTITY= 35  your profit was  $39



COMMAND  : P 15 Q 45 S


Period   3 has been simulated...

With PRICE= 15  and QUANTITY= 45  your profit was  $36



COMMAND  : P 10 Q 35 S


Period   4 has been simulated...

With PRICE= 10  and QUANTITY= 35  your profit was  $45
```

```
COMMAND  : P 20 S

Period   5 has been simulated...

With PRICE= 20  and QUANTITY= 35  your profit was  $12



::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::

Please CAREFULLY answer the following three questions:

If 100 other people were playing this game right now,
  how many would be closer to the optimum than you (0-100)?  : 25

How would you rate the "usability" of this program;
  from 1 to 9, where 1=frustrating, 9=convenient (1-9)?  : 7

How would you describe your present attitude
  toward this game; 1=bored, 9=enjoying it (1-9)?  : 9

::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::



COMMAND  :


***** Only available commands are:

PRICE        Set retail price for this period
QUANTITY     Set production quantity for this period
SIMULATE     Simulate this period's results
HISTORY      Provide History Report
ORDERING     Provide Ordered History Report
GRAPH        Provide Summary Graph


COMMAND  : HIST


History Report for most recent 25 periods.

  PERIOD   PRICE     QTY  PROFIT

       1      15      25      35
       2      15      35      39
       3      15      45      36
       4      10      35      45
       5      20      35      12
```

COMMAND : ORDERING

History Report - ordered by Profit.

PERIOD	PRICE	QTY	PROFIT
4	10	35	45
2	15	35	39
3	15	45	36
1	15	25	35
5	20	35	12

COMMAND : G

```
                        Graph of PROFIT/10  vs.  PRICE,QTY
30 X.................................................................
29 X    .    .    .    .    .    .    .    .    .    .    .    .    .
28 X    .    .    .    .    .    .    .    .    .    .    .    .    .
27 X    .    .    .    .    .    .    .    .    .    .    .    .    .
26 X    .    .    .    .    .    .    .    .    .    .    .    .    .
25 X.................................................................
24 X    .    .    .    .    .    .    .    .    .    .    .    .    .
23 X    .    .    .    .    .    .    .    .    .    .    .    .    .
22 X    .    .    .    .    .    .    .    .    .    .    .    .    .
21 X    .    .    .    .    .    .    .    .    .    .    .    .    .
20 X..................................1..............................
19 X    .    .    .    .    .    .    .    .    .    .    .    .    .
18      .    .    .    .    .    .    .    .    .    .    .    .    .
17 P    .    .    .    .    .    .    .    .    .    .    .    .    .
16 R    .    .    .    .    .    .    .    .    .    .    .    .    .
15 I........................3.........3.........3....................
14 C    .    .    .    .    .    .    .    .    .    .    .    .    .
13 E    .    .    .    .    .    .    .    .    .    .    .    .    .
12      .    .    .    .    .    .    .    .    .    .    .    .    .
11 X    .    .    .    .    .    .    .    .    .    .    .    .    .
10 X..................................4..............................
 9 X    .    .    .    .    .    .    .    .    .    .    .    .    .
 8 X    .    .    .    .    .    .    .    .    .    .    .    .    .
 7 X    .    .    .    .    .    .    .    .    .    .    .    .    .
 6 X    .    .    .    .    .    .    .    .    .    .    .    .    .
 5 X.................................................................
 4 X    .    .    .    .    .    .    .    .    .    .    .    .    .
 3 X    .    .    .    .    .    .    .    .    .    .    .    .    .
 2 X    .    .    .    .    .    .    .    .    .    .    .    .    .
 1 X    .    .    .    .    .    .    .    .    .    .    .    .    .
   XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX QUANTITY XXXXXXXXXXXXXXXXXXXXXXXXX

    12345678911111111112222222222333333333344444444445555555555666666
             0123456789012345678901234567890123456789012345
```

Appendix D

PROFIT FUNCTION

A one-quarter portion of the profit function (read in by the computer game program) appears on the next page, with the highest profit in each row underlined. Clearly, the function is simply a "winding mountain ridge." To recreate the entire profit function, simply reflect the matrix on the next page along the left edge and then along the bottom edge, yielding a "four-arm mountain" with the peak at 80. The profit function is, thus, monotone increasing in two dimensions, with one global maximum and no local maxima - yet is still complex enough to keep each participant thinking.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	2	2	2	3
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	2	2	3	4	4
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	2	2	3	4	4	5
0	0	0	0	0	0	0	0	0	0	0	0	1	1	2	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0	0	0	1	1	2	3	4	5	6	6	7	8
0	0	0	0	0	0	0	0	0	1	1	2	2	3	4	5	6	7	8	9	10
0	0	0	0	0	0	0	0	1	1	2	3	3	5	6	7	8	9	10	11	10
0	0	0	0	0	0	0	1	1	2	3	4	5	6	7	9	10	11	12	11	10
0	0	0	0	0	0	1	1	2	3	4	5	7	8	9	11	12	13	12	11	10
0	0	0	0	0	1	1	2	3	4	6	7	9	10	11	13	14	13	12	11	9
0	0	0	0	1	2	2	3	5	6	8	9	11	12	14	15	14	13	11	10	9
0	0	0	1	2	2	4	5	7	8	10	11	13	15	16	15	14	12	11	9	8
0	0	0	1	2	3	4	5	7	9	10	12	14	16	17	16	14	12	10	9	7
0	0	1	2	3	4	6	7	9	11	13	15	17	18	17	16	14	12	10	9	7
0	0	1	2	3	4	6	8	10	12	14	16	18	19	18	16	14	12	10	8	6
0	1	2	3	5	6	8	10	12	14	16	18	20	19	18	16	14	12	10	8	6
0	1	2	3	5	6	9	11	13	15	17	19	21	19	17	15	13	11	9	6	5
0	1	2	3	5	7	9	11	14	16	18	20	22	20	18	16	14	11	9	7	5
1	2	4	5	7	9	12	14	17	19	21	23	22	20	18	16	14	11	9	7	5
1	2	4	6	7	10	12	15	17	20	22	24	22	20	17	15	12	10	7	6	4
1	3	4	6	8	10	13	15	18	21	23	25	23	21	18	15	13	10	8	6	4
3	4	6	8	11	13	16	19	21	24	26	25	23	21	18	15	13	10	8	6	4
3	4	6	8	11	14	17	19	22	25	27	25	22	19	17	14	11	8	6	4	3
3	4	6	9	11	14	17	20	23	26	28	26	23	20	17	14	11	9	6	4	3
3	4	7	9	12	15	18	21	24	27	29	27	24	21	18	15	12	9	7	4	3
5	7	9	12	15	18	22	25	28	30	29	27	24	21	18	15	12	9	7	4	3
5	7	10	13	16	19	22	25	29	31	29	25	22	19	16	13	10	7	5	3	2
5	7	10	13	16	20	23	26	30	32	30	26	23	20	16	13	10	7	5	3	2
5	8	10	14	17	20	24	27	30	33	30	27	24	20	17	14	10	8	5	3	2
5	8	10	14	17	21	24	28	31	34	31	28	24	21	17	14	10	8	5	3	2
5	8	11	14	18	22	25	29	32	35	32	29	25	22	18	14	11	8	5	4	2
9	11	15	19	23	27	30	34	37	36	33	30	26	22	18	15	11	8	6	4	2
9	12	16	20	24	28	32	36	39	36	32	28	24	20	16	12	9	6	4	2	0
9	13	17	21	25	29	34	38	41	38	34	29	25	21	17	13	9	6	4	2	0
10	13	18	22	26	31	35	40	43	40	35	31	26	22	18	13	10	7	4	2	0
14	18	23	28	32	37	42	45	44	41	36	32	27	23	18	14	10	7	5	2	0
14	19	24	29	34	39	43	47	43	39	34	29	24	19	14	11	7	5	2	0	0
15	20	25	30	35	40	45	49	45	40	35	30	25	20	15	11	8	5	3	0	0
21	26	31	37	42	47	51	50	46	41	36	31	26	21	15	12	8	5	3	0	0
22	27	33	38	43	49	53	49	43	38	33	27	22	16	12	8	5	3	0	0	0
23	28	34	39	45	51	55	51	45	39	34	28	23	17	13	8	6	3	0	0	0
30	36	42	48	54	58	56	52	46	40	34	29	23	17	13	9	6	3	0	0	0
31	38	44	50	56	61	56	50	44	38	31	25	19	14	9	6	3	0	0	0	0
39	46	53	59	64	62	57	51	45	38	32	25	19	14	10	6	3	0	0	0	0
41	48	55	62	67	62	55	48	41	34	27	21	15	10	7	3	0	0	0	0	0
50	57	65	70	68	63	56	49	42	35	28	21	16	10	7	3	0	0	0	0	0
60	67	73	71	66	58	51	44	36	29	22	16	11	7	4	0	0	0	0	0	0
70	76	74	68	61	53	46	38	30	23	17	11	8	4	0	0	0	0	0	0	0
80	78	72	64	56	48	40	32	24	18	12	8	4	0	0	0	0	0	0	0	0

Appendix E

SAMPLE PROGRAM OUTPUT

A listing of the computer game output for an individual player appears on the next page. On that page, if a line is preceded by a 0, it is introductory information; if by a 1, it is the results of another period of simulation; if by a 2, it is a set of attitude questionnaire outcomes; and if by a 3, it is termination information. The labels at the bottom of the output refer to the lines preceded by a 1. The PERIOD, PRICE, QTY, PROFIT, #HISTORYS, #ORDERS, #GRAPHS, and SECS*10 (time) labels should be obvious. The other lines are:

#GETLITS - # of string literals inputted from the user
#GETLINS - # of times a new input line was typed by the user
#COMMANDS - # of commands executed by the user
#DEFAULTS - # of commands having default responses available
#ERRORS - # of errors mad by the user
MAXCHARS - maximum # of characters the user could have typed
NUMCHARS - actual # of characters the user did type
#GETNUMS - # of numbers inputted from the user
#HELPS - # of times a help message was displayed
#NONADEF - # of alphabetic defaults not accepted by the user
#NONNDEF - # of numeric defaults not accepted by the user

```
ONAME=SAMPLE         MX= 16  MY=  9  RZ=1.196
OMODE=2  (CMD)       PRICE=10  QTY=25  DEF=YES
1  1  1  1  1 9 7 3 0 020 3 2 0 0 0 100 5379
1  2  3  3  2 5 1 3 0 021 3 2 0 0 0 000  342
1  3  5  5  2 5 1 3 0 021 3 2 0 0 0 000  287
1  4  9  9  1 5 1 3 0 021 3 2 0 0 0 000  138
1  5 13 13 15 5 1 3 0 021 3 2 0 0 0 000  626
2 50  5  9
1  6 15 35 60 7 3 4 0 026 4 2 0 0 0 001 1069
1  7 15 40 38 5 1 3 0 021 3 2 0 0 1 000  178
1  8 15 45 60 5 1 3 0 021 3 2 0 0 1 000  241
1  9 15 50 59 5 1 3 0 021 3 2 0 0 1 000  322
1 10 15 30 59 5 1 3 0 021 3 2 0 0 1 000  194
1 11 15 25 45 5 1 3 0 021 3 2 0 0 1 000  190
1 12 15 20 32 5 1 3 0 021 3 2 0 0 1 000  105
1 13 20 35 22 5 1 3 0 021 3 2 0 0 0 000  188
1 14 20 40  4 5 1 3 0 021 3 2 0 0 1 000  134
1 15 10 35 63 6 2 4 0 026 4 2 0 0 0 001  440
2100  7  9
1 16 10 40 86 5 1 3 0 021 3 2 0 0 1 000  130
1 17 10 45 63 5 1 3 0 021 3 2 0 0 1 000  177
1 18  5 40 76 5 2 3 0 021 3 2 0 0 0 000  185
1 19 10 45 63 5 1 3 0 021 3 2 0 0 0 000  191
1 20  8 40 95 7 3 4 0 026 4 2 0 0 0 001  418
3  1 30 40  8 9 1 1 9 9 1

L  P  P  Q  P # # # # # M N # # # # ###     S
I  E  R  T  R G G C D E A U G H N N HOG     E
N  R  I  Y  O E E O E R X M E E O O IRR     C
E  I  C     F T T M F R C C T L N N SDA     S
   O  E     I L L M A O H H N P A N TEP     *
C  D        T I I A U R A A U S D D ORH     1
O             T N N L S R R M   E E RSS     0
D             S S D T   S S S   F F Y
E                 S S           S S S
```

Appendix F

SAMPLE PROTOCOLS

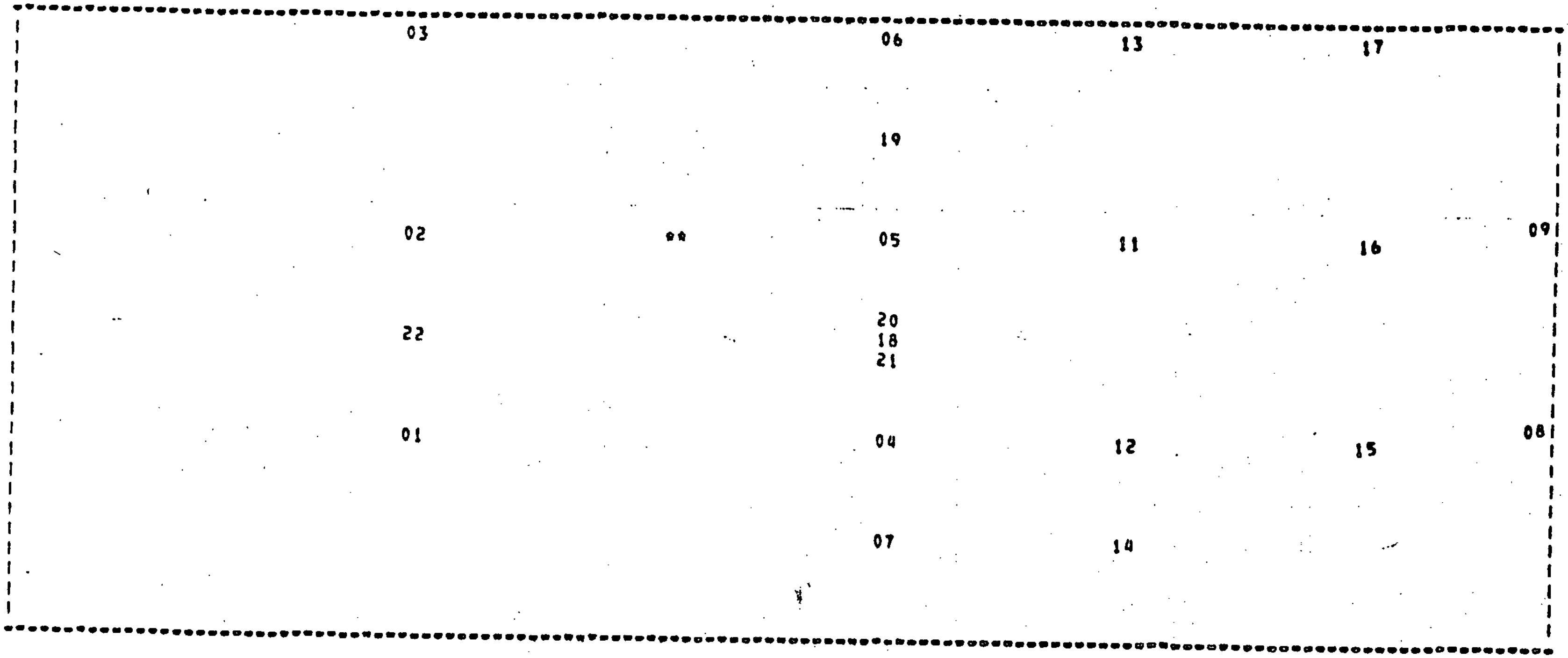
Three examples of user protocol diagrams appear on the next three pages. On theses diagrams, '***' indicates the position of the optimum profit; below the diagram, it is indicated whether or not the subject found the optimum. The 2-digit numbers indicate the order in which <price, quantity> pairs were simulated (imagine price running from 1 to 30 along the vertical axis, and quantity running from 1 to 70 along the horizontal axis). By connecting the points, one can get a good feel for what the original participant was up to (see the end of chapter six for further details).

SAMPLE = "SYSTEMATIC PROTOCOL"

```
--------------------------------------------------------------------------------
!                                                                              !
!                                                                              !
!                 02                        13                                 !
!                                                                              !
!                                           **                                 !
!            05   01                   10 04121109  08                         !
!                                                                              !
!                 03                       07       06                         !
!                                                                              !
!                                                                              !
!                                                                              !
--------------------------------------------------------------------------------
```

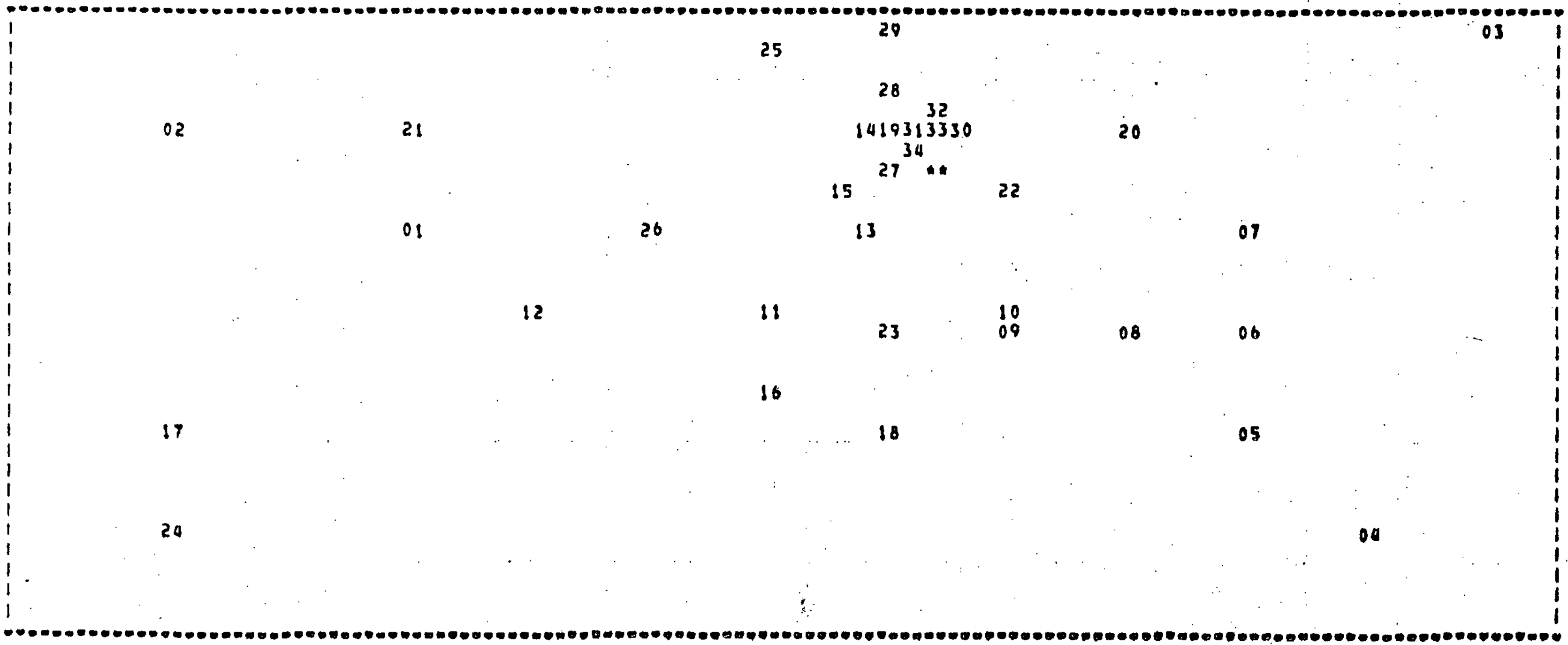
FINISHED

SAMPLE = "STRUCTURED TRIAL AND ERROR"



NOT FINISHED

SAMPLE - "RANDOM SEARCH"



FINISHED

Appendix G

SUMMARY OF RESULTS

This last appendix summarizes the results of the statistical tests of the 26 hypotheses contained in this thesis (see chapter six for details). In the summary on the next pages, the hypotheses are broken down into subparts whenever necessary. In the last column, it is noted whether each hypothesis was rejected or accepted (i.e. supported), based upon whether or not the null hypothesis (of equality) was accepted or rejected, respectively.

HYP.	DEPENDENT VARIABLE	IND. VAR.	SIGN.	A/R
1.	Minutes/Period	Mode	ns	Rej
		Exp	0.09	Acc
		Style	ns	Rej
		Risk	0.05	Acc
2.	Termination	Mode	ns	Rej
		Exp	0.00	Acc
		Style	ns	Rej
		Risk	ns	Rej
3.	Confidence	Mode	ns	Rej
		Exp	0.01	Acc
		Style	0.08	Acc
		Risk	0.11	Acc
4.	Game Version	Min/Per.	ns	Rej
		Term.	ns	Rej
		Confid.	ns	Rej
5.	Experience Level	Min/Per.	0.04	Acc
		Term.	0.00	Acc
		Confid.	0.00	Acc
6.	Cognitive Style	Min/Per.	0.15	Acc
		Term.	0.05	Acc
		Confid.	0.01	Acc
7.	Risk Attitude	Min/Per.	0.02	Acc
		Term.	0.11	Acc
		Confid.	0.03	Acc
8.	Error Rate	Mode/Exp	ns	Rej
9.	Opening Defaults	-----	---	Acc
10.	YES/NO Defaults	-----	ns	Acc
11.	Acc. of Defaults	Exp	ns	Rej
		Style	0.05	Acc
		Risk	ns	Rej
12.	Extent of Abbrev.	Mode	0.01	Acc
		Exp	ns	Rej
		Style	ns	Rej
		Risk	0.04	Acc
13.	Abbreviation	Length	0.00	Acc
14.	Comp. over Time	Min/Per.	0.00	Acc
15.	Comp. over Time	Confid.	ns	Rej
16.	Comp. over Time	Usability	0.09	Acc
17.	Comp. over Time	Abbrev.	0.01	Acc

18.	Ccmp. over Time	Histories	0.00	Acc
19.	Comp. over Time	Ord-Hist	ns	Rej
20.	Comp. over Time	Graphs	0.02	Acc
21.	History Reports	Mode	0.00	Acc
		Exp	0.03	Acc
		Style	ns	Rej
		Risk	ns	Rej
22.	Ordered Hist. Reports	Mode	0.01	Acc
		Exp	ns	Rej
		Style	ns	Rej
		Risk	ns	Rej
23.	Graphs	Mode	ns	Rej
		Exp	0.11	Acc
		Style	0.15	Acc
		Risk	ns	Rej
24.	Protocol Structure	Exp	0.13	Acc
		Style	0.03	Acc
		Risk	ns	Rej
25.	Protocol Dispersion	Exp	0.07	Acc
		Style	ns	Rej
		Risk	ns	Rej