THE MODELING OF HUMAN INTELLIGENCE IN THE COMPUTER AS
DEMONSTRATED IN THE GAME OF DIPLOMAT

by
James Edward Collins

and
Thomas Dean Paulsen

June 1970

This document has been approved for public release and sale; its distribution is unlimited.
The Modeling of Human Intelligence in the Computer

As Demonstrated in the Game of DIPLOMAT

by

James Edward Collins
Lieutenant Commander, United States Navy
B. S., United States Naval Academy, 1959

and

Thomas Dean Paulsen
Lieutenant Commander, United States Navy
B. S., United States Naval Academy, 1960

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
June 1970

Author

James E. Collins

Author

Thomas D. Paulsen

Approved by:

Robert C. Bolles
Thesis Advisor

Chairman, Committee for Computer Science

R. J. Ruchti
Academic Dean
ABSTRACT

The purpose of this thesis is a discussion of developing human-like behavior in the computer. A theory of the human learning processes is first described. This leads to the presentation of a computer game which simulates the human capabilities of reasoning and learning. The program is required to make intelligent decisions based on past experiences and critical analysis of the present situation.
# TABLE OF CONTENTS

## I. INTRODUCTION

- I. INTRODUCTION ............................................. 9

## II. THEORY

- II. THEORY .................................................. 10
  - B. REPRESENTATION OF THE HUMAN LEARNING PROCESS .. 13
  - C. BACKGROUND OF THE GAME OF DIPLOMAT ............ 14

## III. THE GAME

- III. THE GAME ............................................... 20
  - A. THE REASONING PROCESSES .............................. 28
    1. Subroutine STRTGY .................................... 28
    2. Subroutine CPNDCN .................................... 34
  - B. MEMORY STRUCTURE AND THE LEARNING PROCESS ...... 37
    1. The Memory Structure ................................ 37
    2. The Learning Process ................................ 40
  - C. ANALYSIS OF THE GAME .................................. 44

## IV. CONCLUSIONS

- IV. CONCLUSIONS ............................................. 46
  - A. EXTENSIONS AND CHANGES ............................... 46
  - B. POSSIBLE USES OF DIPLOMAT ........................... 48

## APPENDIX A

- APPENDIX A - THE MEMORY STRUCTURE OF DIPLOMAT ....... 50

## APPENDIX B

- APPENDIX B - EXAMPLE COMPUTER TERMINAL OUTPUT ...... 52

## APPENDIX C

- APPENDIX C - COMPUTER PROGRAM LISTING ............... 74

## BIBLIOGRAPHY

- BIBLIOGRAPHY ................................................. 160

## INITIAL DISTRIBUTION LIST

- INITIAL DISTRIBUTION LIST .............................. 162

## FORM DD 1473

- FORM DD 1473 ................................................. 163
LIST OF TABLES

TABLE

I. Wealth Changes Versus Strategies
   As a Function of Player Strength ............. 21
LIST OF FIGURES

FIGURE

1. Macro-Flowchart of the Game DIPLOMAT .............. 24
2. Approximate Payoff Matrix Presented to Player (SOUTH) ........................................ 25
3. Macro-Flowchart of Subroutine STRTGY Indicating the Determination of the Desired Function ............................... 29
4. Basic Algorithm of STRTGY .............................. 30
5. Basic Algorithm of CPNDCN .............................. 35
6. Category Values of Strategies Used in Pattern-Matching ........................................... 42
7. Prediction of the Next Move Based on Strategy Type ............................................... 43
I. INTRODUCTION

In the past few years, much attention in the computer world has been given to the study and development of Artificial Intelligence. One goal of artificial intelligence is to develop human-like qualities in the computer. One method of patterning human behavior is achieved by giving the computer the ability to (1) absorb data, (2) use inductive reasoning to make generalizations based on the data and abstract information from the data, (3) make decisions based on these abstractions and generalizations, and (4) learn from past experience. Admittedly this seems to be an enormous task, and in fact it is, yet this is how a child learns. To go one step further in modeling human behavior, and at the same time provide a reasonable limit on the size of the computer required, the ability to "forget" long-past experience could be developed in the system.

The theory of the human learning process is discussed first in this thesis. The theory leads to the development of a computer program that learns based on data gathered from past experiences. The program uses its library of stored knowledge in making decisions. The example which is presented is a game named DIPLOMAT which simulates the representatives of two nations interacting in strategic negotiations regarding relative strengths, i.e., armaments. The game is played between two opponents, the computer and a person.

The authors of the thesis consider that the interaction of the computer and a human in a decision-making environment is an excellent way of demonstrating the ability of the computer to develop human-like behavior.
II. THEORY

One of the objectives of artificial intelligence is to build a computer system that will effectively mimic the human learning process. This would be an enormous task if the system were to be so general in nature that it could handle any conceivable task, because it would involve representing the external world in a form adaptable for the computer. When restricted to a small subset of the world, i.e., confined to one or two specific tasks, attempts at this objective have been fairly successful in that the systems do mimic and sometimes surpass the human in their performance in these areas. However when trying to model the human brain in the general sense, the problem has been found to be extremely complex.

A possible approach to the problem would be to model the human brain using the physiological approach, i.e., to build the exact electrical network of sensors, storages, and connectors that form the physical makeup of the brain. Obviously this is impractical because the brain is composed of so many cells. Dr. R. L. Beurle, a noted English authority on artificial intelligence [Ref. 3] who has extensively studied the theory of brain models, estimates that the brain is composed of approximately $10^{10}$ neurons (nerve cells).

Another approach to the problem might be the psychological approach. In this approach, it is necessary to model the logical structure of the brain rather than the physical structure. However in order to use this approach, some knowledge of the process of psychological development of human behavior must be obtained. It would be best to trace the learning processes from their beginning in a child, since a knowledge of
total human behavior usually requires almost a lifetime of study and experience.

In the area of the development of the mental abilities of the child, the works of one man stand out above all others - the works of Jean Piaget. Jean Piaget is a noted Swiss child psychologist who was educated at Hevchotel, Zurich, and the University of Paris. He has been a professor of child psychology and history of scientific thought at the University of Geneva since 1929, and is the director of the International Bureau of Education at the Institute J. J. Rousseau. Above all, he is noted for his research in the development of the mind from birth to adolescence.

The works of Jean Piaget, as presented in References 9 through 11, form the basis of this section on the theory of the development of the human learning process. They are supported by References 15 and 16, which are basic references used in the American medical profession.

A. THE LEARNING PROCESS OF THE HUMAN MIND

Piaget contends that the human mind consists of a finite number of structures, each consisting of a finite number of cells. Each of these cells contains an element of information which is a part of the human thought process. He theorizes that there is a separate set of information structures for each function of the central nervous system which is composed of the brain and spinal cord. Many of these separate sets of structures are developed before birth, for example, a set controlling the action of the heart, another the function of breathing, a third controlling the flexing of the arms and legs. Piaget further contends that as relationships build, these structures are linked together to produce automatic reflexes and in general the structures are reordered
and enlarged in successive phases. The central theme for the human learning process is then the process of manipulation of the information structures, i.e., the building, storing, linking, rebuilding and re-linking of elements of information.

It is easiest to describe the manipulating of these information structures by studying the development of the brain of the newborn baby. This is the approach presented by Piaget. At birth the cerebral cortex, which is composed principally of neurons, is largely undeveloped, and the infant is basically a reflex organism. The reflexes are the result of the manipulation of the information structures before birth. Other than the regular actions of the heart and other vital systems, most muscle activity is random, lacking direction from the brain.

One of the first learning processes of the infant is the learning of spatial relationships. While flexing his arms and legs, he touches the side of the crib. Relationships are built in his mind between the information structures for the sense of touch and those of muscle control, and a new structure built up regarding an awareness of the confining walls of the crib. These structures become libraries of information. Other libraries build up, for example he grows to associate his mother with warmth, comfort, and nourishment. The building of relationships between the structures, and the reordering of the structures and links, forms an associative memory within the brain.

It is important in the study of the human brain to understand that these activities are the result of relationships between the libraries even though the information in their structures is dissimilar. Once the links are established, the dissimilar sets function together. The act of crying which is the result of combining the structures of muscle control,
an awareness of a specific need, respiration control, and others is a good example. The operation that is accomplished constitutes some action of uniting or separating, placing or displacing, arranging or disarranging elements of information into sets of structures. As the relationships develop within the brain, the mind learns how to use the data which has been structured within its memory to produce a desired result, such as rolling over or sitting up.

B. REPRESENTATION OF THE HUMAN LEARNING PROCESS

The process of thought is the result of the human becoming aware of the relationships that link certain structures in the associative memory of his brain. Piaget explains the mechanism of thought as a movement which evolves when an awareness of the relationships becomes sufficiently advanced to permit the individual to combine the information from several structures into a single idea. The resulting thought may cause other reactions such as body movement, and thus may cause new structures to be created or new links to be built. The actual process involved in the human brain in conceiving a thought is not definitely known, and must be extremely complicated. If such a scheme is to be computerized, however, it must be made deterministic.

In an attempt to make the process a deterministic one, it is necessary to summarize some facts deduced from the discussion of Piaget's works. One result of the thought process is the acquiring of an idea, which is the outcome of the interaction of cells or structures. The combination of old structures in a new way may lead to new and possibly improved concepts. The new relationships may be formed by applying deductive reasoning to some information extracted from the interconnected structures. In this way, the learning of a new concept may be achieved.
Thus learning is not merely an additive process, i.e., the piling of one disjoint piece of information atop another and another and another.

The number of cells or structures is not the criteria for learning, rather it is the effective combining of the stored information that results in intelligence. Once a certain point is reached in the process of human development, the physical size of the brain does not rapidly become larger and larger, rather information is restructured, old information may be forgotten and new information stored in its place, and new links established.

These two techniques, structuring of memory and learning by interaction of the structured information, are utilized in the structuring of a computer memory and the development of a program which will exhibit deductive reasoning based on learning. This is the example presented in this thesis. Considerable research and soul searching went into formulating a worthwhile application for the model of human intelligence. The application had to be general enough to apply to real-life situations, yet not overly complicated.

C. BACKGROUND OF THE GAME OF DIPLOMAT

Several recent periodicals have been devoting considerable attention to games as an application of artificial intelligence. One game in particular has been the subject of much of the attention, the game known as the Prisoner's Dilemma. The classic prisoner's dilemma is described in reference 7 as follows:

"Two suspects are taken into custody and separated. The district attorney is certain that they are guilty of a specific crime, but he does not have adequate evidence to convict them at a trial. He points out to each prisoner that each has two alternatives: to confess to the crime the police are sure they have done, or not to confess. If they both do not confess, then
the district attorney states he will book them on some minor trumped-up charge such as petty larceny and illegal possession of a weapon, and they will both receive minor punishment; if they both confess they will be prosecuted, but he will recommend less than the most severe sentence; but if one confesses and the other does not, then the confessor will receive lenient treatment for turning state's evidence, whereas the latter will get 'the book' slapped at him."

One reason the prisoner's dilemma has been discussed so much is that there are numerous situations in the world that have some of the characteristics of this game (or extensions from this game such as the addition of participants and/or strategies). Most economic situations that require a choice among a finite number of strategies have these characteristics. Consider, for example, gasoline service stations located close to one another, each of which can lower its prices. Regardless of the price one's competitors set, any one manager is better off, in the short run at least, cutting his price. If all cut prices, however, the total volume of business is the same as if none cut prices, but the total revenue is less. On a larger scale, consider wheat farmers in a country without governmental price and production controls. Any one farmer is better off producing wheat as long as his marginal cost is not greater than the price. He will be able to sell all he can produce at the going market rate without affecting the price. If all farmers produce maximum amounts, however, the price will be pushed down and all will be worse off than if each had restricted his production. On the worldwide scale, there is the problem of disarmament. One country can be more powerful (or secure) by arming, but nothing is gained if all arm. All countries would be better off if all disarmed in that the money not spent on defense could be spent for, say, consumer goods or for correcting social ills.

More interesting than the one-time classic prisoner's dilemma is the iterated game, i.e., a game composed of many moves. In the overall
picture of the iterated game, each player must (in general) forsake the possibility of maximizing his own short-run profit to enjoy the greatest payoff by maximizing his long-run profit. With a one-trial game and an unknown rival, it is difficult to imagine the wisdom of choosing a medium-gain, little-risk, cooperating type strategy, when more can be gained (or lost) by choosing the high-risk high-payoff strategy (nothing ventured, nothing gained). The single trial situation eliminates both the possibility of future cooperation and the possibility of punishing a rival for non-cooperative action in one trial. Dr. Lester B. Lave of the Carnegie Institute of Technology [Ref. 6] has studied factors affecting cooperation in prisoner's dilemma type games. He has found that the single-trial game and the multi-trial game are basically equivalent in the formal sense in that the expected values of the two games have the same range across different groups of opponents. However, the games are quite different with respect to negotiating cooperation among different participants. The expected values of the two games are not equal for a given rival, since certain forms of behavior can induce cooperation or competition. He based these results on experiments conducted using human competitors only, and did not introduce computer gaming into his research. He further found through experimentation that when a game was iterated, it was possible to display behavior that induces or stifles cooperation. He also found that it was possible for the players to develop communication between them using the choices in the game, but that this rudimentary form of communication took time to establish and function. He found that the longer the game, the more likely it was that a stable cooperative solution could be achieved.

Other experiments with a complex decision task showed that experience from previous tasks was a large factor in success. The conclusions
drawn from studies conducted at the Human Performance Center, Ohio State University, [Ref. 13] were that after gaining experience when tested under realistic circumstances, the experienced subjects were in general less conservative than naive subjects who received no such prior training. They were willing to take more risks to achieve higher overall gains.

The study of concept attainment by the machine has also been the subject of study. In order for a human to play a game of this nature, he must be able to form a few concepts of the game itself and of his opponent. It can be likened to the game of poker in that opponents must deduce the type of individuals playing the game. Dr. Frank B. Baker, a professor of educational psychology at the University of Wisconsin, has studied the theory of concept attainment and developed a computer program which demonstrated the theory in a simple decision task requiring the identification of common attributes among different sets [Ref. 1]. In Reference 2, Dr. Baker states:

"If computer programs are to serve as useful models of cognitive behavior, their creators cannot avoid coming to grips with the necessity for establishing an internal organization for their model which implements the higher level cognitive behavior associated with the human capacity for self-direction, autocriticism, and adaptation."

In computer game playing, the concept of the game itself is built into the game, however the concept of different methods of play or types of strategies that the computer may face is something that must be attained as it plays.

Most of the behavioral tests conducted to date have been between humans. However, in the past few years, more attention has been devoted to simulating these tests on the computer using interaction between man and the machine. Professor Roman J. Weil of the University of Chicago stated in Reference 18 the advantage of using the computer for this application quite well:
"The philosophy underlying the computer approach is this: If a program can be constructed that, when placed in a prisoner's dilemma situation, exhibits behavior like the behavior of people when placed in the same situation, then that program will be a powerful tool for generalizations."

Professor Weil goes on to say that if the computer can be made to simulate the human in organizing data and making decisions under all simulated conditions of risk and stress, it will be possible to accumulate more data and more accurately predict human behavior in the same environment.

The game of DIPLOMAT presented in this thesis is basically an extended version of the prisoner's dilemma. It incorporates an iterated game technique with the game lasting anywhere from ten to fifty moves, and two opponents choosing from among three strategies. Additional complications to the prisoner's dilemma basis are inserted by varying the payoffs to the participants as a function of previous moves, and by inserting an unknown random variance into the payoff table.

In order to be successful in this game, the participants must perform most of the tasks listed in the introduction as a goal of artificial intelligence, in addition to performing the task of concept attainment. In particular, the computer must analyze the situation of the game at the time of the move and refer to past games and past moves in the present game to determine its opponent's probable strategy. It must then abstract enough information from his prior and present knowledge to select a strategy, and correctly analyze the results of the move in order to store (remember) meaningful experience. In addition, it must form concepts regarding the reliability or honesty of its opponents. The participants in this game may or may not be completely truthful in their
negotiations, which is certainly characteristic of actual diplomats at
the conference table. The computer, then, must form estimates of its
opponent's reliability and factor this into its selection of a strategy.

In the first game played, the computer has no prior knowledge upon
which to draw, and so must reason from an analysis of the present situa-
tion. With each move however, the system acquires more experience and
thus has a better base from which to draw in selecting strategies. When
the computer plays its second game, it has the experience of the first,
with one winning and one losing strategy, to use for reference. In
general, the more games it has played, the more experienced it is and
the better it performs in making important decisions.
III. THE GAME

DIPLOMAT models the representatives of two nations interacting in strategic negotiations regarding armaments. It is basically a non-zero-sum two person rectangular game, using the phraseology of formal game theory. Each nation is given three choices of possible strategies regarding armaments:

- Strategy 1: Increase Armaments (Arm)
- Strategy 2: Maintain the Status Quo
- Strategy 3: Decrease Armaments (Disarm)

The following basic concepts govern the decision of the strategy to be followed by each side:

- Each nation starts the game with zero strength and zero wealth, where "zero" implies a deviation from the average rather than absolute zero.

- Arming increases strength by one unit, disarming decreases strength by one unit, and maintaining the status quo does not change strength. Arming costs money decreasing wealth, disarming gains wealth, and maintaining the status quo may increase or decrease wealth, depending on the strength of the player: If the player is strong in armaments, it will cost him more for upkeep and maintenance and hence decrease wealth; if the player is weak, armament upkeep is low and maintaining the status quo should gain some wealth. The basic changes in wealth for the different strategies are shown as a function of strength of the player in Table I.
Changes to the basic values in Table I are generated by a random integer amount between the values of -2 and 42. These changes, which are to simulate economic conditions, are generated at random times during the game; hence, a given economic condition may last for only one move or for many moves. The economic condition for each player may be different, as each opponent uses a different random number generated from the same random number seed. The economic condition in effect for each player is not furnished to either opponent, but must be estimated based on the results of each move. The economic conditions of -2 and -1 mimic those times when prices are high, and the conditions of +1 and +2 mimic those times when the cost of living is relatively low. These values are added to the basic wealth change values of Table I to determine the actual wealth changes for the strategies chosen.

Point values for each participant are determined after each move according to the formula:

\[
\text{POINTS} = (\text{Previous}) + 2 \times (\text{Wealth}) + 5 \times (\text{Strength})
\]

<table>
<thead>
<tr>
<th>STRATEGY</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S &lt; -6$</td>
<td>0</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>$-6 \leq S &lt; -4$</td>
<td>0</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>$-4 \leq S &lt; -2$</td>
<td>-1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>$-2 \leq S &lt; +2$</td>
<td>-2</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>$+2 \leq S &lt; +4$</td>
<td>-3</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>$+4 \leq S &lt; +6$</td>
<td>-4</td>
<td>-2</td>
<td>3</td>
</tr>
<tr>
<td>$S &gt; +6$</td>
<td>-6</td>
<td>-3</td>
<td>2</td>
</tr>
</tbody>
</table>

Table I
Wealth Changes Versus Strategies As a Function of Player Strength
Bonus points are given for the following combinations of strategies:

If both opponents cooperate in disarming, each receives two bonus points as a reward for their cooperation.

If both opponents arm, each receives -1 bonus point, because both have spent money without acquiring any relative advantage.

If one side arms and the other disarms, the opponent who arms is awarded four bonus points for "outfoxing" the other.

The length of the game is at least ten and at most fifty moves. Between these values, a random selection is used for the decision to end the game; as the number of moves increases, the greater the chance of random termination. Experience has indicated that the average length of the game is twenty moves.

The winner of the game is decided by one of three methods:

Normal Termination: The participant with the most POINTS at the end of the game is declared the winner.

Abnormal Termination: The game may be abnormally terminated, even before ten moves have been completed, in two ways. The game is stopped if one nation's wealth becomes thirty units greater than the other, and the richer nation is declared the winner. Similarly, if one participant becomes stronger in armaments than the other by ten units, the game is stopped and the stronger nation is declared the winner.

In playing the game of DIPLOMAT, each side initially declares a proposed strategy called that participant's Concession Point, with each side taking turns declaring the first concession point. The proposal of each player is used as an aid in deciding that player's probable actual strategy. After both concession points have been declared, the computer's
move is locked into the system and the human participant is asked to declare his final strategy for that move. Of course, the concession point and the strategy need not be the same, but wisdom must be used in selecting proposals versus actual strategies in order to maintain a high degree of reliability, yet keep the opponent off guard as to the actual strategy to be chosen.

The program is written as a main routine which controls the running of the game itself, and forty-one subroutines. Seven of the subroutines assist in controlling the game and in performing list-processing chores, nine of the subroutines assist the program in accomplishing its reasoning capabilities, and the other twenty-five subroutines are necessary for accomplishing the task of learning. Among the tasks performed by these forty-one subroutines are setting up the memory, saving and restoring experience, selecting a strategy and a concession point, determining the opponent's reliability, pattern matching data from previous games and moves to take advantage of prior experience, and so forth. The program is written in the FORTRAN IV language, and is designed to operate on-line on a computer terminal using the Cambridge Monitor System. A complete listing of the program is contained in Appendix C.

Figure 1 is a macro-flowchart of the game. DIPLOMAT progresses in the following manner:

After initializing counters, point values, and payoff tables, the system sets up the memory cells in an associative memory structure, filling in the data from previous games. The initial economic conditions are also determined.

The player (hereafter called SOUTH) signs into the system with his name. The computer program (hereafter called NORTH) then
Figure 1. Macro Flowchart of the Game DIPLOMAT
checks to see if he has played this opponent before, and if so brings his history to the top of the catalog of players.

The player inputs a random integer number for use as a seed in generating random numbers throughout the program. The computer simulates flipping a coin to see who will make the first proposal during the first move. SOUTH calls the flip of the coin. (In the discussion which follows, it will be assumed that SOUTH won the toss).

Having won the toss of the coin, SOUTH must make the first concession point. To aid him in making his selection of a proposal and ultimately of a strategy, a payoff matrix is presented to him showing the approximate payoffs to SOUTH (relative to NORTH) for the various combinations of strategies. Approximate payoffs are shown rather than actual because neither SOUTH nor NORTH have access to the existing economic conditions. A sample payoff matrix as presented to SOUTH is shown in Figure 2.

```
SOUTH

STRATEGY       1          2          3
N   1          * -1          1          2
O   *           *           0          1
R   2          * 1          0          1
T   *           *           *          2
H   3          * 2          -i         2

Figure 2
Approximate Payoff Matrix Presented to Player (SOUTH)
```

The approximate payoff matrix changes according to the strengths of NORTH and SOUTH because of the differences in wealth for the various strategies as shown in Table I.
SOUTH may propose a concession point of 1 (arming), 2 (maintaining the status quo), or 3 (disarming). NORTH then considers SOUTH's concession point, attempts to determine if SOUTH is being honest, analyzes his payoff matrix, and responds to SOUTH with his concession point. NORTH also decides upon a strategy at this time.

SOUTH considers NORTH's concession point and decides upon a strategy for the move. Both sides then enter their strategies into the system and the results are tabulated and presented for analysis by both participants.

For the next move, NORTH will declare his concession point first. The game proceeds in this manner, alternating between NORTH and SOUTH as to who is first to declare a concession point. After each move, each side analyzes the results in order to determine the economic conditions in effect. After ten moves have been completed and after each move thereafter, a random number is generated and tested to determine if the game should end.

At the end of each move, changes to the payoff tables generated as a result of changing strengths of the participants are calculated and inserted into the system, and it is determined if it is time to change the economic conditions. If so, they are calculated and inserted into the game.

A sample output of the program as exhibited at the counter terminal is located in Appendix B.

In playing this game, the computer program maintains data regarding past moves and past games in order to draw upon this experience in selecting strategies and concession points in future moves and in future
games. It thus forms concepts of each player as it proceeds. At the end of each move, some of the learning subroutines are called upon to update the short-term memory in order to store data for use in playing the game in progress and for maintaining running totals. At the end of the game, others of these subroutines calculate the game totals and determine the characteristics exhibited by both participants for inclusion in the long-term memory. This is the experience gained by the program from this game.

The data which forms the experience is kept in an associative memory structure mainly for ease of manipulation, but it is considered that this patterns the human in organizing data and experiences in his mind. In studies conducted of neural nets, it has been found that the human will take data, organize it into logical structures based on determining relationships between units of the data, and store it in an associative net accordingly [Ref.s 2 and 3].

Diagrams of the memory structure of DIPLOMAT are contained in Appendix A, and were conceived by the authors after critical analysis of the structure of data maintained about players during hand simulation of the game.

Besides exhibiting the human attribute of learning by storing away past experiences, the program mimics the human in its reasoning capability in deciding upon strategies and concession points. The method of reasoning was also patterned after analysis of the mental reasoning used during hand simulation of the game. These two attributes of reasoning and learning are discussed in greater detail in the sections which follow.
A. THE REASONING PROCESSES

Two of the subroutines of the program are designed to mimic the reasoning processes of the human. These are the strategy decision subroutine (STRTGY) and the concession point decision subroutine (CPNDCH). These subroutines in turn call on many other subroutines to determine optimum strategies, probable moves of the opponent, next moves of the computer, and so forth. Both the strategy and concession point decision subroutines were written based on the thought processes used by the authors in playing the game by hand.

1. Subroutine STRTGY

   This subroutine has three main functions. The first is that of strategy decision based on an analysis of all the factors available. This decision is final if the player’s concession point is known, otherwise it is a tentative decision until the declaration of SOUTH’s concession point. The second function is that of reconsideration of the tentative strategy after SOUTH has declared his concession point, and results in the final selection of a strategy. This function is called upon only if NORTH was first to declare a concession point. The third function is analysis of the completed move in order to determine if a better choice of strategy could have been made. If so, an adjustment of the calculations performed in the decision portion of the routine is accomplished.

   Macro-flowcharts of STRTGY are shown in Figures 3 and 4. The subroutine may be called upon either two or three times each move, depending upon who submits the first proposal. A series of flags are used to determine for which function the subroutine is being called. The algorithm for making this determination is shown in Figure 3.
Figure 3. Macro Flowchart of Subroutine STRTG1 Indicating Determination of the Desired Function
Figure 4. Basic Algorithm of STRTGY
Figure 4 provides the basic algorithm of STRATEGY. The decision upon a strategy for the move is based upon the consideration of four factors, each of which are weighted in a "polynomial" fashion. These four terms of the polynomial are based upon:

1) SOUTH's concession point,
2) POINTS relative to SOUTH,
3) Previous experience, and
4) Economic conditions.

The coefficients assigned to each of these terms are inserted into the system at the start of the game, and may become modified as the game progresses, as discussed in the description of the analysis function.

Even before considering any of the four factors, the computer checks to ensure that it is not in danger of losing because of relative wealth or strength disadvantage. If it is, the system immediately selects as its strategy the one which will gain the most wealth or strength, as needed, and returns to the main program without considering any of the four factors. If the computer finds that this emergency action is necessary, later analysis of the move is not performed.

If emergency action is not necessary, each of the four factors are considered and used in determining a final strategy for the move.

The first factor to be considered is SOUTH's concession point. The computer first determines if it wants to call for a renegotiation. Renegotiation is requested if it finds that its opponent's proposal was to arm and that he is quite strong already; it requires SOUTH to submit a new concession point, although the new proposal may be the same as the original. The next step after considering (and possibly carrying out) renegotiation is to attempt to determine SOUTH's probable forthcoming strategy. The computer selects candidates for its opponent's
strategy based on weighted reliability estimates, patterns of dishonesty in SOUTH's proposals, history of previous games with this player, and formal game theory. After deciding upon SOUTH's probable strategy, the computer selects as the first term of the polynomial the strategy which maximizes NORTH's gain relative to SOUTH if SOUTH actually selects that strategy.

The second factor to be considered in selecting a strategy is based on NORTH's points relative to SOUTH. Depending upon whether the computer is behind, ahead, or even with its opponent, this term of the polynomial is set to the strategy which maximizes NORTH's possible gain or minimizes its possible loss. Formal game theory is used in selecting these possible strategies.

The third factor to be considered is the most difficult one, because it is based upon previous experience. If SOUTH has been a previous opponent (within the last ten opponents), there exists a history of his previous games in the computer's long term memory. NORTH can use this information in predicting SOUTH's probable moves. In addition, NORTH can search other strategy types contained in its libraries of past games in order to select previously successful strategies to use against its opponent. After several moves have been completed, the computer searches all the strategy listings in its "experience" in order to classify SOUTH's general pattern of strategies. Each of these listings may contain pointers to other strategy patterns which have proven successful against SOUTH's pattern in the past. NORTH can also pattern-match its own strategy pattern against those existing in the library in order to determine its own predicted move. The computer occasionally selects a
strategy opposite to the predicted one to avoid being stereotyped by SOUTH. This is called a "guess-opposite" selection of a strategy. It is the third factor of the polynomial which gains from the learning capabilities of the machine. The method of learning and pattern-matching is discussed in greater detail in a later section.

The fourth and final factor to be considered in selecting a strategy is based on an estimate of the economic condition in effect. After the completion of each move, the program analyzes the results to determine if they match the expected values. If not, it estimates the economic conditions and sets this term of the polynomial to the strategy which takes the most advantage of the state of the economy. For example, if prices are low, it is probably the best time to arm, but if prices are abnormally high it may be too expensive to arm at that time.

After all four factors have been considered in selecting a strategy, the system conducts a vote to determine the choice for the move. If SOUTH's concession point has not been declared, the coefficient of the first term is set to zero, and a tentative strategy is chosen based on the other three terms. When it becomes time for the reconsideration, the vote is taken of all four terms for deciding the final strategy. The vote is accomplished by summing the coefficients of the terms voting for each of the three possible strategies. The strategy which receives the greatest sum is the "winning" strategy; in case of a tie, the computer selects from the tieing strategies the one which will result in the greatest absolute point gain.

Having selected a strategy, control is returned to the main program which either calls upon Subroutine CI..DCN to decide NORTH's concession point, or else locks the strategy into the system awaiting SOUTH's indication of a strategy.
The third function of Subroutine STRTGY is that of analyzing the completed move in order to determine if a better strategy could have been chosen. The system does this by entering NORTH's relative payoff table with the actual strategy chosen by SOUTH to determine the strategy which provides the maximum relative gain. If the computer determines that the correct strategy was chosen, no adjustments are performed and control is returned to the main program. If, however, the best strategy was not picked by NORTH, the computer determines if any of the votes cast for the different strategies matched the best possible strategy. If any are found, the coefficients corresponding to those terms are then increased (tuned up), and those assigned to the wrong terms tuned down. If no votes are found, the analysis was unsuccessful for that move.

If the computer ultimately wins the game, the final values of the coefficients for each of the terms are inserted into the long-term memory of the player for use as the initial values in the succeeding game with that player. This is done because the system found these coefficients successful and they would probably provide a better base from which to start the next time. This is but another part of the learning process of the system.

Thus, Subroutine STRTGY attempts to determine a strategy to be followed in the game of DIPLOMAT in much the same way that a human reasons through the game. Similar to a human, if the reasoning process results in a wrong answer, the system attempts to improve itself.

2. Subroutine CPNDCN

This is the second of the program's subroutines which employs human-like reasoning. Its purpose is to determine a concession point to be proposed to SOUTH after having selected a tentative (or final) strategy. Figure 5 provides a macro-flowchart of this subroutine.

34
Figure 5. Basic Algorithm of CPND CN
NORTH's concession point must be carefully chosen. In order to keep the player off guard, the computer must try to maintain at least a facade of reliability, yet not be so reliable as to be "read like a book." If NORTH and SOUTH are each following a pattern of moves, the computer should indicate that it expects to follow the pattern, even if it has no such intention.

For the initial moves, the computer uses a semi-random pattern of selecting proposals. Weight is given to choosing either the actual strategy proposed, or to choosing a passive (i.e., disarming or status quo) declaration. After the initial moves have been completed, however, the computer utilizes much of the data it has been accumulating on the player during the game. In addition to pattern-matching previous moves, the computer maintains estimates of the player's reliability as well as its own. EN is NORTH's estimate of SOUTH's honesty, and ES is what NORTH thinks SOUTH estimates for the computer's reliability. These are both weighted values, with the most weight given to the recent moves. EN and ES are calculated after each move by comparing the actual strategy with the concession point. The values of EN and ES drop accordingly each time there is not a match.

If ES is high, is increasing, or at least is greater than EN, the system determines its predicted move from pattern-matching its own strategies, and uses a predicted move as its concession point. If, on the other hand, NORTH's reliability needs improving, the subroutine normally selects the actual strategy determined by STRTGY as the computer's concession point.

As with STRTGY, CPNDCN was written based on hand simulation of the game. Notes were kept on the reasons behind each decision, and
used to help determine the methods of reasoning developed in these
subroutines.

B. MEMORY STRUCTURE AND THE LEARNING PROCESS

The information within this section is somewhat more detailed be-
cause it describes the attempts to exemplify the concepts of Piaget in
structuring the mind. Computer list-processing techniques are used in
structuring and manipulating the memory for this task. The reason for
this is that these techniques model the theoretical structures of the
brain and are efficient in coding requirements.

1. The Memory Structure

The memory structure of the program, depicted in Appendix A,
is divided into two segments, the "temporary" or short-term memory and
the "permanent" or long-term memory. The short-term memory contains
data regarding the current game and the long-term memory contains the
data which comprises the experience gained from previous games.

The short-term memory is used to maintain data necessary for
determining the pattern of play being used by both the human player and
the computer. The information within this segment of memory grows as
the game progresses. At the end of the game, pertinent data is summarized
and transferred to the long-term memory, and the short-term memory is
deleted or "forgotten." The temporary memory consists of arrays and
lists for determining the reliability estimates used in the reasoning
processes of the system. In addition, this segment of memory contains
a pattern of strategies used by each participant. It also maintains a
total of the number of times each strategy is selected. The former is
used in generally classifying the method of play which is being employed,
and the latter is used in determining the overall aggressiveness being exhibited.

The long-term memory, the "experience," consists of arrays and lists arranged into four libraries. Libraries are interconnected as they are developed, in order to maintain continuity in the flow of information and facilitate access to specific data. These four libraries are labeled the "Catalogue of Players," the "Library of Initial Moves," the "Library of Types of Moves," and the "Library of Sequences of Moves." For simplicity these names are abbreviated the "Catalogue," the "Initial Library," the "Type Library," and the "Sequence Library."

To provide a basic understanding of the construction and maintenance of these libraries, it is necessary to define various terms associated with them. A move is a played strategy. Groups of three moves are categorized into five levels of aggressiveness. Groups are thus combinations of any three strategies played. The following example illustrates the construction of groups:

Suppose an opponent picked the following moves (strategies):

\[ 1 \ 2 \ 1 \ 1 \ 3 \ 2 \ 2 \ 1 \ 3 \ 3 \ 1 \ 1 \ 1 \ 2 \ 1 \ 3 \]

Group (1) then consists of 1 2 1, Group (2) is 2 1 1, Group (3) is 1 1 3, Group (4) is 1 3 2, and so on.

Patterns are determined in the Initial Library according to individual moves, and are determined in the Type Library by a sequence of groups. The sequence of groups is called a "TYPE." Further discussion will be given to the utilization of categories of aggressiveness, and their associated groups, in the explanation of the learning process of the program.
The Catalogue contains information on one to ten players. As a player is first introduced into the game, a unique set of nodes (sequential set of computer words) is established for him which at the end of the game will be filled with data and pointers regarding his history of play. The most recent player is maintained at the top of the catalogue and the oldest player at the bottom. As players participate in additional games, a reordering of the Catalogue is accomplished to maintain this perspective. If the Catalogue reaches its capacity of ten players and a new player is introduced into the system, the information on the oldest player is deleted (forgotten) and the new player established in the Catalogue.

Information maintained on each player includes pointers to the initial strategies and a sequence of "TYPE" patterns used by both the player and the computer in previous games. It also contains the set of parameters (the foundation of the coefficients considered in the strategy polynomial) which have proven most successful against that player, the average values of reliability demonstrated by both the player and the computer when playing against that player, and the player's aggressiveness in previous games.

The INIT Library contains the initial three moves (first group) used in various games in the past, and pointers to the Type Library. Information on the exact initial strategies used in previous games is maintained so that the computer can try to get a "jump" on the player at the beginning of the game. It is used to predict SOUTH's initial strategy and to aid the computer in selecting the best offense against this strategy.
The Type Library contains up to ten structures. A structure consists of a header cell followed by a TYPE of eight to ten groups. The header cell contains pointers to other structures in the Type Library which have demonstrated a successful defense against this TYPE, and against which this TYPE has proven to be a good offense.

The Sequence Library is a history of the TYPES and associated initial strategies for each game played by a particular player in the Catalogue. It also contains the computer's TYPES used against any player in the Catalogue. Each entry in the Sequence Library contains a pointer to a structure in the Type Library and a pointer to the initial strategy associated with that structure.

When the capacity of each of these libraries is filled, the oldest information is deleted, or updated, to make room for the new information. This conforms to the contention that the brain grows to a finite size, then information becomes restructured or forgotten as it grows out of date.

2. The Learning Process

The learning process progresses as the memory structures develop. During the first game the computer plays, there is nothing in long-term memory so the computer must reason through the game as best it can. During succeeding games, however, the computer searches long-term memory for strategy predictions and pattern-matching types of strategies.

When the player signs into the system at the start of a game, the computer searches the Catalogue to determine if this player has been played before. If so, the computer assumes that he will follow
the same initial strategy and determines the best moves against that strategy. After two moves, the computer pattern-matches these against the INIT Library to find the closest match. The computer's third move is selected as the one which is best against the third move listed in the closest matched INIT group. After the third move, the computer again pattern-matches the initial three moves against the INIT Library, finds the closest match, and predicts that the player will use the TYPE to which that entry in the INIT Library points.

The Type Library provides general categories of strategies to simplify pattern-matching. In a twenty-move game, there could be an almost infinite number of sequences of exact strategies used, but by categorizing exact strategies into general classifications, pattern-matching may be done at a meta-level. These categories, as stated earlier, are determined by arranging the moves into groups of three and classifying the group into one of five levels of aggressiveness. Figures 6 and 7 illustrate the techniques of categorizing strategies and making predictions based on the types obtained. The overlap of the groups provides continuity in classifying and predicting.

As the game progresses, the computer pattern-matches the TYPES used by both the player and the computer, which are maintained in short-term memory, against the Type Library. This provides NORTH with predicted moves for both participants, and also with an indication of the TYPE that is best to use against SOUTH's TYPE.

The Sequence Library provides the computer with the ability to make better predictions of the TYPES that a player will use. After each game with a player, a pointer to the TYPE employed in that game is
<table>
<thead>
<tr>
<th>CATEGORY</th>
<th>1</th>
<th>A1</th>
<th>2</th>
<th>A2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Combination of Groups of Three Moves</td>
<td>111</td>
<td>113</td>
<td>123</td>
<td>133</td>
<td>332</td>
</tr>
<tr>
<td></td>
<td>112</td>
<td>131</td>
<td>132</td>
<td>331</td>
<td>323</td>
</tr>
<tr>
<td></td>
<td>121</td>
<td>311</td>
<td>213</td>
<td>313</td>
<td>233</td>
</tr>
<tr>
<td></td>
<td>211</td>
<td>122</td>
<td>231</td>
<td>322</td>
<td>333</td>
</tr>
<tr>
<td></td>
<td>212</td>
<td>312</td>
<td>232</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>221</td>
<td>321</td>
<td>223</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>222</td>
</tr>
</tbody>
</table>

The sum of the strategies for each category is:

- Category 1 - 3 or 4
- Category Al - 5
- Category 2 - 6
- Category A2 - 7
- Category 3 - 8 or 9

Figure 6. Category Values of Strategies Used in Pattern-Matching
By totaling the first two moves of any predicted category, it is possible to predict the next move:

<table>
<thead>
<tr>
<th>If First Two Moves Total:</th>
<th>For Category Number:</th>
<th>The Move Predicted Is:</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2 1 A1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>A1</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>A1</td>
<td>1</td>
</tr>
<tr>
<td>Overlapping of Category Groups Shown Above</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Example Strategies and Categories are Inserted</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>A2</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>A2</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>A2</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>2</td>
</tr>
</tbody>
</table>

Figure 7. Prediction of the Next Move Based on Strategy Type
inserted in the Sequence Library. The computer can pattern-match these lists to determine the expected type which the player will use during the next game.

At the end of a game, corresponding data between the short-term memory and the long-term memory are compared. This is accomplished by the same methods of pattern-matching used during the game. If similarity exists, the structures of the short-term memory are combined with the closest Initial and Type Library structures to form more up-to-date information. If no similarities exist, the information from temporary memory is transferred to permanent memory as new Initial and Type Library entries. If these libraries are at capacity, the oldest information is deleted (forgotten) and the new information is inserted.

As suggested by the theory, past experience is utilized to obtain the best prediction of events, but if the computer finds no matching experience upon which to draw, it must reason through the problem as best it can. Knowledge grows when new structures are formed or old ones reconstructed.

C. ANALYSIS OF THE GAME

After a rather poor start, the computer has gone ahead of its opponents in total points for all games, and the gap is widening. This is due partly to some minor changes in the strategy decision routines, however evidence indicates that most of the credit can be given to the building of the libraries. This contention is supported by the fact that the analysis portion of the strategy routine tends to increase (tune up) the basis of the coefficient assigned to the experience factor, and tune down some of the others.
Based on observation of the game and discussions with those playing it, it is evident that the players and the computer use many of the same factors in deciding upon a strategy and a concession point. In addition, concepts built by the human are similar to the concepts formed by the computer and stored within its memory. The average values for the reliability estimates that the computer maintains on both itself and its opponent remain relatively constant as the game progresses. The same is true of the weighted reliability.
IV. CONCLUSIONS

"If a program can be constructed that, when placed in a prisoner's dilemma situation, exhibits behavior like the behavior of people when placed in the same situation, then that program will be a powerful tool for generalizations."

The above statement by Professor Weil is repeated from an earlier section of this thesis for emphasis. In its existing form, this program does provide a powerful tool for extracting generalizations regarding human behavior in a medium risk decision making task. With but minor changes in wording or by incorporating extensions to the game, the program can be made applicable to almost any field of corporate or governmental endeavor requiring a psychological understanding of human behavior in making decisions where different gains can be achieved at varying risks.

A. EXTENSIONS AND CHANGES

DIPLOMAT is itself an extension to the classic prisoner's dilemma game. It may be extended or changed to broaden its applicability to real world situations and make the game considerably more interesting.

The simplest change to the program would be to change the name assigned to the three strategies. For example, the applicability of the program could be changed by transforming the words arm/status-quo/disarm into the words buy/wait/sell, or perhaps raise-prices/no-change/lower-prices.

The whole outcome of the game can be changed drastically by changing
the POINTS formula. This is done by simply assigning different co-
efficients to the factors of WEALTH and STRENGTH, or by changing the
values in Table I (the wealth changes for the various strategies shown
as a function of player strength). Thus the game may be altered to
match actual conditions encountered in, say, the business community.

The number of strategies from which to choose could be changed in
either direction. Decreasing the choice to two would render the game
closer to the prisoner's dilemma situation, but even this has many ap-
plications in the real world. On the other hand, it would be more in-
teresting to increase the number of strategies. For example, there
could be two levels of aiming and two levels of disarming, or for buying
or selling. Some implications of increasing the number of strategies,
however, would be that the pattern-matching routines may not be feasible
in their present state. It would probably require a pattern-matching
scheme which placed more emphasis on the meta-level, i.e., looking at the
broad spectrum of the pattern from a higher level, rather than pattern-
matching individual strategies or small groups of strategies.

Increasing the number of players is probably the most difficult of
the possible extensions to the game, but provides the most interesting
possibilities. If the number of players is increased, treaties between
the players and alliances among groups of players may be proposed and
formed. An infinite number of situations may arise out of this idea, for
example, one nation might wonder whether his ally will abide by the al-
liance or possibly turn on him several moves hence; or, a player might
hesitate to sign an agreement when a better one might be offered from a
different player. In the diplomacy situation, several nations may dis-
arm to gain wealth, then form an alliance against a stronger nation which
has armed to gain strength at the cost of wealth. Increasing the number of players would also make the concepts of reasoning and learning more difficult, but more fascinating. For example, will one nation risk an alliance with another when it remembers prior treachery? Instead of merely analyzing what one's opponent thinks of him, a player will have to analyze several players' estimates of all the opponents. The reliability considerations become almost overwhelming.

B. POSSIBLE USES OF DIPLOMAT

It is considered that Professor Weil was correct in assuming that the prisoner's dilemma computer game would be a powerful tool for generalizations. The game of DIPLOMAT or its extensions would be an invaluable aid in the training of executives prior to stepping into positions requiring the art and finesse of personal contact. The speed of the computer permits the game to proceed rapidly and permits many different situations to be established by changing payoffs and formulas. The performance of different players against standard setups could then be analyzed.

In addition, personnel in the study of behavioral science and psychology could develop a better understanding of the nature of the human decision-making process and of the risk of striking out on an independent path as opposed to the benefits and security gained by cooperating. The fact that the risky path may lead to greater gains may be more important to some people than to others.

The training of college students in the theory of marketing and analysis would also be enhanced by applying textbook concepts to the difficulties of the everchanging conditions and types of people with
which they may deal. This training might prove useful for young officers in the diplomatic corps or even at junior level armed forces colleges.

In order to enhance the development of computer application to this type of work, it is recommended that consideration be given to the possibility of adapting this program to marketing or financial decision-making courses in the management curriculum. This type of program could be extremely useful as a tool in teaching management students the power and usefulness of the computer. It demonstrates the interface capabilities of man and machine. It can also be used as a teaching aid in artificial intelligence, game theory, and basic management-decision courses. The first part of Section III describing the game could be reproduced for use as a handout for potential players.

The concept and performance of DIPLOMAT appears to be good. It is ahead of the human players in POINTS and performs well in adapting to different human strategies that have been tried against it. However, similar to a human, it does not win every game, indicating that even it has more to learn.
Short-Term Memory
DO YOU WISH TO READ MEMORY AND REBUILD THE HISTORY OF PLAYERS, YES OR NO "A3" FORMAT.
YES

THIS PROGRAM Requires the following inputs:
1. YOUR NAME IN A4 FORMAT, RIGHT JUSTIFIED IF LESS THAN FOUR
   CHARACTERS IN LENGTH.
2. INRAND, AN ODD INTEGER OF NINE OR LESS DIGITS, OR RANDOM ORIGIN,
   FOR USE IN GENERATING RANDOM NUMBERS.

YOUR NAME:
JIM

INRAND:
555

A TOSS OF A COIN WILL BE USED TO DETERMINE WHO GOES FIRST.
YOU MAY CALL HEAD OR TAIL:
HEAD

SORRY YOU LOST THE TOSS OF THE COIN.
THEREFORE I WILL MAKE THE FIRST PROPOSAL (CONCESSION POINT).

PARAMETERS AT START OF GAME ARE: 10  10  10  20
MY PROPOSAL (CONCESSION POINT) WHERE 1=ARM, 2=MAINTAIN THE STATUS QUO, AND 3=DISARM IS: 2.

THE APPROXIMATE PAYOFF MATRIX FOR THE EXISTING CONDITIONS OF STRENGTH AND WEALTH IS SHOWN BELOW. INDICATED PAYOFFS ARE TO SOUTH (YOU).

<table>
<thead>
<tr>
<th>STRATEGY</th>
<th>SOUTH</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td></td>
<td>-1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>O</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R</td>
<td></td>
<td>-1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>T</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H</td>
<td></td>
<td>2</td>
<td>-1</td>
<td>2</td>
</tr>
</tbody>
</table>

WHAT IS YOUR PROPOSAL, STRATEGY 1(ARM), 2(MAINTAIN THE STATUS QUO), OR 3(DISARM): ?

IT IS NOW TIME TO CARRY OUT THE STRATEGY OF EACH SIDE. THE COMPUTER'S MOVE IS LOCKED INTO THE SYSTEM. PLEASE INDICATE YOUR STRATEGY AS 1, 2, OR 3:

THE ACTUAL STRATEGIES CHOSEN BY EACH OPPONENT ARE AS FOLLOWS:

NORTH(COMPUTER): 1
SOUTH(PLAYER): 3

THE RESULTS OF THE LAST MOVE ARE AS FOLLOWS:

<table>
<thead>
<tr>
<th>WEALTH</th>
<th>STRENGTH</th>
<th>POINTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMPUTER(NORTH)</td>
<td>-1</td>
<td>1</td>
</tr>
<tr>
<td>PLAYER(SOUTH)</td>
<td>5</td>
<td>-1</td>
</tr>
</tbody>
</table>
IT IS NOW MOVE NUMBER 2. IT IS YOUR TURN TO GO FIRST.

THE APPROXIMATE PAYOFF MATRIX IS UNCHANGED FROM THE LAST MOVE. YOU MAY USE IT AS AN AID IN DECIDING YOUR STRATEGY AND CP.

WHAT IS YOUR PROPOSAL, STRATEGY 1(ARM), 2(MAINTAIN THE STATUS QUO), OR 3(DISARM)?

MY PROPOSAL (CONCESSION POINT) WHERE 1=ARM, 2=MAINTAIN THE STATUS QUO, AND 3=DISARM IS: 1.

NOTE THAT MY CP IS TO ARM.

PLEASE INDICATE WITH A "YES" OR "NO" (RIGHT JUSTIFIED IN A3 FORMAT) IF YOU WISH TO RENEGOTIATE:

NO

IT IS NOW TIME TO CARRY OUT THE STRATEGY OF EACH SIDE.

THE COMPUTER'S MOVE IS LOCKED INTO THE SYSTEM.

PLEASE INDICATE YOUR STRATEGY AS 1, 2, OR 3:

1

THE ACTUAL STRATEGIES CHOSEN BY EACH OPPONENT ARE AS FOLLOWS:

NORTH(COMPUTER): 1
SOUTH(PLAYER): 1

THE RESULTS OF THE LAST MOVE ARE AS FOLLOWS:

<table>
<thead>
<tr>
<th>WEALTH</th>
<th>STRENGTH</th>
<th>POINTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMPUTER (NORTH)</td>
<td>-2</td>
<td>2</td>
</tr>
<tr>
<td>PLAYER (SOUTH)</td>
<td>4</td>
<td>0</td>
</tr>
</tbody>
</table>
IT IS NOW MOVE NUMBER 3. IT IS MY TURN TO GO FIRST.

MY CP IS: 2.

THE APPROXIMATE PAYOFF MATRIX IS UNCHANGED FROM THE LAST MOVE. YOU MAY USE IT AS AN AID IN DECIDING YOUR STRATEGY AND CP.

WHAT IS YOUR PROPOSAL, (1, 2, OR 3)?

2

WHAT IS YOUR STRATEGY?

2

THE ACTUAL STRATEGIES CHosen BY EACH OPPONENT ARE AS FOLLOWS:

NORTH (COMPUTER): 1
SOUTH (PLAYER): 2

THE RESULTS OF THE LAST MOVE ARE AS FOLLOWS:

<table>
<thead>
<tr>
<th>WEALTH</th>
<th>STRENGTH</th>
<th>POINTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMPUTER (NORTH)</td>
<td>-6</td>
<td>2</td>
</tr>
<tr>
<td>PLAYER (SOUTH)</td>
<td>7</td>
<td>0</td>
</tr>
</tbody>
</table>
IT IS NOW MOVE NUMBER 4. IT IS YOUR TURN TO GO FIRST.

THE APPROXIMATE PAYOFF MATRIX FOR THE EXISTING CONDITIONS OF STRENGTH AND WEALTH IS SHOWN BELOW. INDICATED PAYOFFS ARE TO SOUTH (YOU).

<table>
<thead>
<tr>
<th>STRATEGY</th>
<th>SOUTH</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>N</td>
<td>1</td>
</tr>
<tr>
<td>O</td>
<td>2</td>
</tr>
<tr>
<td>T</td>
<td>3</td>
</tr>
</tbody>
</table>

WHAT IS YOUR PROPOSAL, (1,2, OR 3)?

3

MY CP IS: 2.

WHAT IS YOUR STRATEGY?

THE ACTUAL STRATEGIES CHOSEN BY EACH OPPONENT ARE AS FOLLOWS:

NORTH(COMPUTER): 1
SOUTH(PLAYER): 1

THE RESULTS OF THE LAST MOVE ARE AS FOLLOWS:

<table>
<thead>
<tr>
<th>WEALTH</th>
<th>STRENGTH</th>
<th>POINTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMPUTER (NORTH)</td>
<td>-11</td>
<td>* -4</td>
</tr>
<tr>
<td>PLAYER (SOUTH)</td>
<td>* -5</td>
<td>* 1</td>
</tr>
</tbody>
</table>
IT IS NOW MOVE NUMBER 5. IT IS MY TURN TO GO FIRST.

MY CP IS: 3.

THE APPROXIMATE PAYOFF MATRIX IS UNCHANGED FROM THE LAST MOVE. YOU MAY USE IT AS AN AID IN DECIDING YOUR STRATEGY AND CP.

WHAT IS YOUR PROPOSAL, (1, 2, OR 3)?

2

WHAT IS YOUR STRATEGY?

THE ACTUAL STRATEGIES CHOSEN BY EACH OPPONENT ARE AS FOLLOWS:

NORTH(COMPUTER): 3
SOUTH PLAYER): 1

THE RESULTS OF THE LAST MOVE ARE AS FOLLOWS:

<table>
<thead>
<tr>
<th>WEALTH</th>
<th>STRATEGY</th>
<th>POINTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMPUTER (NORTH)</td>
<td>-2</td>
<td>3</td>
</tr>
<tr>
<td>PLAYER (SOUTH)</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>
IT IS NOW MOVE NUMBER 6. IT IS YOUR TURN TO GO FIRST.

THE APPROXIMATE PAYOFF MATRIX IS UNCHANGED FROM THE LAST MOVE.
YOU MAY USE IT AS AN AID IN DECIDING YOUR STRATEGY AND CP.

WHAT IS YOUR PROPOSAL, (1,2,OR 3)?
2
MY CP IS: 3.

WHAT IS YOUR STRATEGY?
3
THE ACTUAL STRATEGIES CHOSEN BY EACH OPPONENT ARE AS FOLLOWS:
NORTH(COMPUTER): 3
SOUTH(PLAYER): 3

THE RESULTS OF THE LAST MOVE ARE AS FOLLOWS:

<table>
<thead>
<tr>
<th></th>
<th>WEALTH</th>
<th>STRENGTH</th>
<th>POINTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMPUTER (NORTH)</td>
<td>-7</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>PLAYER (SOUTH)</td>
<td>3</td>
<td>1</td>
<td>15</td>
</tr>
</tbody>
</table>
IT IS NOW MOVE NUMBER 7. IT IS MY TURN TO GO FIRST.

MY CP IS: 2.

THE APPROXIMATE PAYOFF MATRIX FOR THE EXISTING CONDITIONS OF STRENGTH AND WEALTH IS SHOWN BELOW. INDICATED PAYOFFS ARE TO SOUTH (YOU).

<table>
<thead>
<tr>
<th>STRATEGY</th>
<th>SOUTH 1</th>
<th>SOUTH 2</th>
<th>SOUTH 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>NORTH</td>
<td>**</td>
<td>* -1</td>
<td>1 2</td>
</tr>
<tr>
<td>SOUTH</td>
<td>**</td>
<td>-1 0</td>
<td>1</td>
</tr>
<tr>
<td>HEART</td>
<td>**</td>
<td>2 -1</td>
<td>2</td>
</tr>
</tbody>
</table>

WHAT IS YOUR PROPOSAL, (1, 2, OR 3)?

2

WHAT IS YOUR STRATEGY?

1

THE ACTUAL STRATEGIES CHOSEN BY EACH OPPONENT ARE AS FOLLOWS:

NORTH (COMPUTER): 3
SOUTH (PLAYER): 1

THE RESULTS OF THE LAST MOVE ARE AS FOLLOWS:

<table>
<thead>
<tr>
<th>COMPUTER (NORTH)</th>
<th>WEALTH</th>
<th>STRENGTH</th>
<th>POINTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLAYER (SOUTH)</td>
<td>**</td>
<td>*</td>
<td>*</td>
</tr>
</tbody>
</table>

-5 -1      1 16
IT IS NOW MOVE NUMBER 8. IT IS YOUR TURN TO GO FIRST.

THE APPROXIMATE PAYOFF MATRIX FOR THE EXISTING CONDITIONS OF STRENGTH AND WEALTH IS SHOWN BELOW. INDICATED PAYOFFS ARE TO SOUTH (YOU).

<table>
<thead>
<tr>
<th>STRATEGY</th>
<th>SOUTH 1</th>
<th>SOUTH 2</th>
<th>SOUTH 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>N 1</td>
<td>*</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>OR 2</td>
<td>*</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>T 3</td>
<td>*</td>
<td>4</td>
<td>1</td>
</tr>
</tbody>
</table>

WHAT IS YOUR PROPOSAL, (1, 2, or 3)?

My CP is: 2.

WHAT IS YOUR STRATEGY?

THE ACTUAL STRATEGIES CHOSEN BY EACH OPPONENT ARE AS FOLLOWS:

NORTH (COMPUTER): 3
SOUTH (PLAYER): 2

THE RESULTS OF THE LAST MOVE ARE AS FOLLOWS:

<table>
<thead>
<tr>
<th></th>
<th>WEALTH</th>
<th>STRENGTH</th>
<th>POINTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMPUTER (NORTH)</td>
<td>*</td>
<td>-3</td>
<td>*</td>
</tr>
<tr>
<td>PLAYER (SOUTH)</td>
<td>*</td>
<td>-2</td>
<td>2</td>
</tr>
</tbody>
</table>
IT IS NOW MOVE NUMBER 9. IT IS MY TURN TO GO FIRST.

MY CP IS: 1.

NOTE THAT MY CP IS TO ARM.

PLEASE INDICATE WITH A "YES" OR "NO" (RIGHT JUSTIFIED IN A2 FORMAT) IF YOU WISH TO RENEGOTIATE:

YES

BASED ON YOUR REQUEST FOR RENEGOTIATION, I HAVE CONSIDERED MY CONCESSION POINT. MY NEW CP IS TO: 2.

THE APPROXIMATE PAYOFF MATRIX FOR THE EXISTING CONDITIONS OF STRENGTH AND WEALTH IS SHOWN BELOW. INDICATED PAYOFFS ARE TO SOUTH (YOU).

<table>
<thead>
<tr>
<th>STRATEGY</th>
<th>SOUTH</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>NORTH</td>
<td>*</td>
</tr>
<tr>
<td>SOUTH</td>
<td>*</td>
</tr>
<tr>
<td>H</td>
<td>*</td>
</tr>
</tbody>
</table>

WHAT IS YOUR PROPOSAL, (1, 2, OR 3)?

WHAT IS YOUR STRATEGY?

THE ACTUAL STRATEGIES CHOSEN BY EACH OPPONENT ARE AS FOLLOWS:

NORTH (COMPUTER): 3
SOUTH (PLAYER): 3

THE RESULTS OF THE LAST MOVE ARE AS FOLLOWS:

| WEALTH | STRENGTH | *POINTS*
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>COMPUTER (NORTH)</td>
<td>* 0 *</td>
<td>-1 *</td>
</tr>
<tr>
<td>PLAYER (SOUTH)</td>
<td>* 0 *</td>
<td>-1 *</td>
</tr>
</tbody>
</table>
IT IS NOW MOVE NUMBER 10. IT IS YOUR TURN TO GO FIRST.

THE APPROXIMATE PAYOFF MATRIX IS UNCHANGED FROM THE LAST MOVE. YOU MAY USE IT AS AN AID IN DECIDING YOUR STRATEGY AND CP.

WHAT IS YOUR PROPOSAL, (1, 2, OR 3)?

3

MY CP IS: 3.

WHAT IS YOUR STRATEGY?

2

THE ACTUAL STRATEGIES CHOSEN BY EACH OPPONENT ARE AS FOLLOWS:

NORTH(COMPILER): 3
SOUTH(PLAYER): 2

THE RESULTS OF THE LAST MOVE ARE AS FOLLOWS:

<table>
<thead>
<tr>
<th></th>
<th>WEALTH</th>
<th>STRENGTH</th>
<th><em>POINTS</em></th>
</tr>
</thead>
<tbody>
<tr>
<td>COMPUTER (NORTH)</td>
<td>3</td>
<td>-2</td>
<td>2</td>
</tr>
<tr>
<td>PLAYER (SOUTH)</td>
<td>0</td>
<td>1</td>
<td>15</td>
</tr>
</tbody>
</table>
IT IS NOW MOVE NUMBER 11. IT IS MY TURN TO GO FIRST.

MY CP IS: 2.

THE APPROXIMATE PAYOFF MATRIX FOR THE EXISTING CONDITIONS OF STRENGTH AND WEALTH IS SHOWN BELOW. INDICATED PAYOFFS ARE TO SOUTH (YOU).

<table>
<thead>
<tr>
<th>STRATEGY</th>
<th>SOUTH 1</th>
<th>SOUTH 2</th>
<th>SOUTH 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>NORTH</td>
<td>*</td>
<td>-1</td>
<td>1</td>
</tr>
<tr>
<td>NORTH</td>
<td>*</td>
<td>-1</td>
<td>0</td>
</tr>
<tr>
<td>NORTH</td>
<td>*</td>
<td>2</td>
<td>-1</td>
</tr>
</tbody>
</table>

WHAT IS YOUR PROPOSAL, (1, 2, OR 3)?

3

WHAT IS YOUR STRATEGY?

THE ACTUAL STRATEGIES CHOSEN BY EACH OPPONENT ARE AS FOLLOWS:

NORTH (COMPUTER): 3
SOUTH (PLAYER): 3

THE RESULTS OF THE LAST MOVE ARE AS FOLLOWS:

<table>
<thead>
<tr>
<th>COMPUTER (NORTH)</th>
<th>WEALTH</th>
<th>STRENGTH</th>
<th>POINTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLAYER (SOUTH)</td>
<td>5</td>
<td>-3</td>
<td>18</td>
</tr>
</tbody>
</table>

3 3 3
IT IS NOW MOVE NUMBER 12. IT IS YOUR TURN TO GO FIRST.

THE APPROXIMATE PAYOFF MATRIX FOR THE EXISTING CONDITIONS OF STRENGTH AND WEALTH IS SHOWN BELOW.

INDICATED PAYOFFS ARE TO SOUTH (YOU).

<table>
<thead>
<tr>
<th>STRATEGY</th>
<th>SOUTH</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>NORTH</td>
<td>*</td>
</tr>
<tr>
<td>SOUTH</td>
<td>*</td>
</tr>
<tr>
<td>H</td>
<td>*</td>
</tr>
</tbody>
</table>

WHAT IS YOUR PROPOSAL, (1,2,OR 3) ?

3

MY CP IS: 2.

WHAT IS YOUR STRATEGY?

THE ACTUAL STRATEGIES CHOSEN BY EACH OPPONENT ARE AS FOLLOWS:

<table>
<thead>
<tr>
<th>NORTH (COMPUTER)</th>
<th>SOUTH (PLAYER)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>

THE RESULTS OF THE LAST MOVE ARE AS FOLLOWS:

<table>
<thead>
<tr>
<th>COMPUTER (NORTH)</th>
<th>WEALTH</th>
<th>STRENGTH</th>
<th>POINTS*</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLAYER (SOUTH)</td>
<td>7</td>
<td>-4</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>1</td>
<td>21</td>
</tr>
</tbody>
</table>
IT IS NOW MOVE NUMBER 13. IT IS MY TURN TO GO FIRST.

MY CP IS: 2.

THE APPROXIMATE PAYOFF MATRIX IS UNCHANGED FROM THE LAST MOVE. YOU MAY USE IT AS AN AID IN DECIDING YOUR STRATEGY AND CP.

WHAT IS YOUR PROPOSAL, (1,2,OR 3)?
2

WHAT IS YOUR STRATEGY?
3

THE ACTUAL STRATEGIES CHOSEN BY EACH OPPONENT ARE AS FOLLOWS:
NORTH(COMPUTER): 3
SOUTH(PLAYER): 3

THE RESULTS OF THE LAST MOVE ARE AS FOLLOWS:

<table>
<thead>
<tr>
<th>COMPUTER (NORTH)</th>
<th>WEALTH</th>
<th>STRENGTH</th>
<th>POINTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>PLAYER (SOUTH)</td>
<td>3</td>
<td>0</td>
<td>24</td>
</tr>
</tbody>
</table>
IT IS NOW MOVE NUMBER 14. IT IS YOUR TURN TO GO FIRST.

THE APPROXIMATE PAYOFF MATRIX IS UNCHANGED FROM THE LAST MOVE. YOU MAY USE IT AS AN AID IN DECIDING YOUR STRATEGY AND CP.

WHAT IS YOUR PROPOSAL, (1,2, OR 3)?

3

MY CP IS: 2.

WHAT IS YOUR STRATEGY?

2

THE ACTUAL STRATEGIES CHosen BY EACH OPPONENT ARE AS FOLLOWS:

NORTH(COMPUTER): 2
SOUTH(PLAYER): 2

THE RESULTS OF THE LAST MOVE ARE AS FOLLOWS:

<table>
<thead>
<tr>
<th></th>
<th>WEALTH</th>
<th>STRENGTH</th>
<th>POINTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMPUTER (NORTH)</td>
<td>12</td>
<td>-5</td>
<td>11</td>
</tr>
<tr>
<td>PLAYER (SOUTH)</td>
<td>13</td>
<td>0</td>
<td>24</td>
</tr>
</tbody>
</table>
IT IS NOW MOVE NUMBER 15. IT IS MY TURN TO GO FIRST.

MY CP IS: 3.

THE APPROXIMATE PAYOFF MATRIX FOR THE EXISTING CONDITIONS OF STRENGTH AND WEALTH IS SHOWN BELOW. INDICATED PAYOFFS ARE TO SOUTH (YOU).

<table>
<thead>
<tr>
<th>STRATEGY</th>
<th>NORTH</th>
<th>SOUTH</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>N</td>
<td>1</td>
<td>-5</td>
</tr>
<tr>
<td>S</td>
<td>2</td>
<td>-3</td>
</tr>
<tr>
<td>H</td>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

WHAT IS YOUR PROPOSAL, (1,2,OR 3)?

3

WHAT IS YOUR STRATEGY?

THE ACTUAL STRATEGIES CHOSEN BY EACH OPPONENT ARE AS FOLLOWS:

NORTH (COMPUTER): 1
SOUTH (PLAYER): 3

THE RESULTS OF THE LAST MOVE ARE AS FOLLOWS:

<table>
<thead>
<tr>
<th>WEALTH</th>
<th>STRENGTH</th>
<th>POINTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>NORTH</td>
<td>13</td>
<td>-4</td>
</tr>
<tr>
<td>SOUTH</td>
<td>7</td>
<td>-1</td>
</tr>
</tbody>
</table>
IT IS NOW MOVE NUMBER 16. IT IS YOUR TURN TO GO FIRST.

THE APPROXIMATE PAYOFF MATRIX IS UNCHANGED FROM THE LAST MOVE. YOU MAY USE IT AS AN AID IN DECIDING YOUR STRATEGY AND CP.

WHAT IS YOUR PROPOSAL, (1, 2, OR 3)?
1
MY CP IS: 1.

NOTE THAT MY CP IS TO ARM.
PLEASE INDICATE WITH A "YES" OR "NO" (RIGHT JUSTIFIED IN A3 FORMAT) IF YOU WISH TO RENEGOTIATE:
YES

BASED ON YOUR REQUEST FOR RENEGOTIATION, I HAVE CONSIDERED MY CONCESSION POINT. MY NEW CP IS TO: 2.

WHAT IS YOUR STRATEGY?
2

THE ACTUAL STRATEGIES CHOSEN BY EACH OPPONENT ARE AS FOLLOWS:
NORTH (COMPUTER): 2
SOUTH (PLAYER): 2

THE RESULTS OF THE LAST MOVE ARE AS FOLLOWS:

<table>
<thead>
<tr>
<th></th>
<th>WEALTH</th>
<th>STRENGTH</th>
<th>POINTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMPUTER (NORTH)</td>
<td>16</td>
<td>-4</td>
<td>26</td>
</tr>
<tr>
<td>PLAYER (SOUTH)</td>
<td>16</td>
<td>-1</td>
<td>25</td>
</tr>
</tbody>
</table>
IT IS NOW MOVE NUMBER 17. IT IS MY TURN TO GO FIRST.

MY CP IS: 2.

THE APPROXIMATE PAYOFF MATRIX FOR THE EXISTING CONDITIONS OF STRENGTH AND WEALTH IS SHOWN BELOW. INDICATED PAYOFFS ARE TO SOUTH (YOU).

<table>
<thead>
<tr>
<th>STRATEGY</th>
<th>SOUTH 1</th>
<th>SOUTH 2</th>
<th>SOUTH 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>1</td>
<td>*</td>
<td>-3</td>
</tr>
<tr>
<td>O</td>
<td>2</td>
<td>-3</td>
<td>-2</td>
</tr>
<tr>
<td>H</td>
<td>3</td>
<td>4</td>
<td>1</td>
</tr>
</tbody>
</table>

WHAT IS YOUR PROPOSAL, (1,2,OR 3) ?
1

WHAT IS YOUR STRATEGY?

THE ACTUAL STRATEGIES CHOSEN BY EACH OPPONENT ARE AS FOLLOWS:
NORTH(COMPUTER): 1
SOUTH(PLAYER): 1

THE RESULTS OF THE LAST MOVE ARE AS FOLLOWS:

<table>
<thead>
<tr>
<th>COMPUTER (NORTH)</th>
<th>WEALTH</th>
<th>STRENGTH</th>
<th>POINTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>-3</td>
<td>30</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PLAYER (SOUTH)</th>
<th>WEALTH</th>
<th>STRENGTH</th>
<th>POINTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0</td>
<td>21</td>
<td></td>
</tr>
</tbody>
</table>
IT IS NOW MOVE NUMBER 18. IT IS YOUR TURN TO GO FIRST.

THE APPROXIMATE PAYOFF MATRIX IS UNCHANGED FROM THE LAST MOVE. YOU MAY USE IT AS AN AID IN DECIDING YOUR STRATEGY AND CP.

WHAT IS YOUR PROPOSAL, (1, 2, OR 3) ?

1
MY CP IS: 2.

WHAT IS YOUR STRATEGY ?

THE ACTUAL STRATEGIES CHOSEN BY EACH OPPONENT ARE AS FOLLOWS:

NORTH (COMPUTER): 2
SOUTH (PLAYER): 3

THE RESULTS OF THE LAST MOVE ARE AS FOLLOWS:

<table>
<thead>
<tr>
<th></th>
<th>WEALTH</th>
<th>STRENGTH</th>
<th>POINTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMPUTER (NORTH)</td>
<td>17</td>
<td>-3</td>
<td>32</td>
</tr>
<tr>
<td>PLAYER (SOUTH)</td>
<td>6</td>
<td>-1</td>
<td>24</td>
</tr>
</tbody>
</table>
IT IS NOW MOVE NUMBER 19. IT IS MY TURN TO GO FIRST.

MY CP IS: 2.

THE APPROXIMATE PAYOFF MATRIX IS UNCHANGED FROM THE LAST MOVE. YOU MAY USE IT AS AN AID IN DECIDING YOUR STRATEGY AND CP.

WHAT IS YOUR PROPOSAL, (1, 2, OR 3)?

WHAT IS YOUR STRATEGY?

THE ACTUAL STRATEGIES CHosen BY EACH OPPONENT ARE AS FOLLOWS:

NORTH (COMPUTER): 1
SOUTH (PLAYER): 1

THE RESULTS OF THE LAST MOVE ARE AS FOLLOWS:

<table>
<thead>
<tr>
<th></th>
<th>WEALTH</th>
<th>STRENGTH</th>
<th>POINTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMPUTER (NORTH)</td>
<td>15</td>
<td>-2</td>
<td>32</td>
</tr>
<tr>
<td>PLAYER (SOUTH)</td>
<td>4</td>
<td>0</td>
<td>24</td>
</tr>
</tbody>
</table>
IT IS NOW MOVE NUMBER 20. IT IS YOUR TURN TO GO FIRST.

THE APPROXIMATE PAYOFF MATRIX IS UNCHANGED FROM THE LAST MOVE. YOU MAY USE IT AS AN AID IN DECIDING YOUR STRATEGY AND CP.

WHAT IS YOUR PROPOSAL, (1, 2, OR 3)?
2
MY CP IS: 2.

WHAT IS YOUR STRATEGY?
1

THE ACTUAL STRATEGIES CHOSEN BY EACH OPPONENT ARE AS FOLLOWS:
- NORTH (COMPUTER): 2
- SOUTH (PLAYER): 1

THE RESULTS OF THE LAST MOVE ARE AS FOLLOWS:

<table>
<thead>
<tr>
<th></th>
<th>WEALTH</th>
<th>STRENGTH</th>
<th>POINTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMPUTER (NORTH)</td>
<td>16</td>
<td>-2</td>
<td>34</td>
</tr>
<tr>
<td>PLAYER (SOUTH)</td>
<td>2</td>
<td>1</td>
<td>25</td>
</tr>
</tbody>
</table>
IT IS NOW MOVE NUMBER 21. IT IS MY TURN TO GO FIRST.

MY CP IS: 2.

THE APPROXIMATE PAYOFF MATRIX IS UNCHANGED FROM THE LAST MOVE. YOU MAY USE IT AS AN AID IN DECIDING YOUR STRATEGY AND CP.

WHAT IS YOUR PROPOSAL, (1,2,OR 3)? 2

WHAT IS YOUR STRATEGY? 3

THE ACTUAL STRATEGIES CHOSEN BY EACH OPPONENT ARE AS FOLLOWS:

NORTH(COMPUTER): 3
SOUTH(PLAYER): 3

*** BY A RANDOM SELECTION, THE UMPIRE HAS DECIDED TO END THIS GAME AFTER 21 MOVES. THE PARTICIPANT WITH THE MOST POINTS (POINTS = (2*WEALTH)+(5*STRENGTH)) IS DECLARED THE WINNER.

THE FINAL GRAND TOTALS FOR THIS GAME ARE:

<table>
<thead>
<tr>
<th></th>
<th>WEALTH</th>
<th>STRENGTH</th>
<th>POINTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>NORTH</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SOUTH</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
</tbody>
</table>

THE COMPUTER WON THE GAME. THANK YOU FOR PLAYING.

PARAMETERS AT END OF GAME ARE: 14 6 16 4
* THE MAIN PROGRAM CONTROLS THE OPERATION OF THE GAME OF *
* DIPLOMAT. IT IS SUPPORTED IN THIS TASK BY THE SEVEN *
* SUBROUTINES LISTED WITHIN THIS SECTION. *

IMPLICIT INTEGER (A-W)
COMMON/TDJE/APN,APS,STACK(40)
COMMON/PAR/P(4),S(4),PRBS(5)
COMMON/EEVEN,ES,TOPFN,TOPE
COMMON/ENP/ENARRY(50),ESARRY(50)
COMMON/CLS/DEFES,BOFES,BOFFN,PMOVEN,PMOVES
COMMON/JEC/VALUE(7,3),RTABN(3,3),RTABS(3,3),ATABN(3),ATABS(3).
COMMON/ST/STTRUE,LASTGO,NFLAG,TURN,ANAL,RECON
COMMON/TD/TOPM,ROTM,CATLG,CHAR,HIST,PLAYER,CATLGN,PATRNS,PATRNN
COMMON/TDP2/STK(10),STKM(10),STKN(10),STKNM(10)
COMMON/TDP3/TYPEN,TYPE1,TYSOU,TYCN
COMMON/TDP8/RLHST,NAHIST
COMMON/TDP9/TYPE(20),INIT(20)
COMMON/TDP9/SEG(20),SCONTR
COMMON/TDP9/GROUP,NSP,NGROUP,PMOVES,TMOVEN,MCNT,CONTR
DIMENSION COMPN(3,3)
DIMENSION RLINK(1000),LLINK(1000),DOWN(1000),RLINK(1000),DOWN
11(1000),DOWN2(1000)
EQUIVALENC (NAME4(4),RLINK(3),LLINK(2),DOWN4(1)),(NAME2(2),DOWN
12(1)),(NAME3(3),RLINK(3),DOWN3(1))
DATA HEADS/'HEAD'/TAILS/'TAIL'/YES/'YES'/NO/'NO'/
DATA A1/'A1'/A2/'A2'/

APN=0
APS=0
CATLGN=0
NC=1
SCONTR=0
INCN=0
INCNF=10
CONTR=0
DO 120 I=1,4
PRBS(I)=0
120 S(I)=0
PRBS(5)=0
FACTXN=0
FACTXS=0
VALUE(1,1)=0
VALUE(2,1)=0
VALUE(4,1)=1
VALUE(5,1)=2
VALUE(6,1)=3
VALUE(7,1)=4
VALUE(1,2)=5
VALUE(2,2)=6
VALUE(4,2)=7
VALUE(5,2)=8
VALUE(6,2)=9
VALUE(7,2)=10
VALUE(1,3)=11
VALUE(2,3)=12
VALUE(4,3)=13
VALUE(5,3)=14
VALUE(6,3)=15
VALUE(7,3)=16

(INITIALIZING ROUTINE)
CALL INITL
CALL ECALC(EXACT,ESACT)
CALL REMEBR(INCNT,INCNTR)

(INPUTS)
WRITE (6,399)
399 FORMAT (1x,' THIS PROGRAM REQUIRES THE FOLLOWING INPUTS:
1. INNAME, THE NAME OF THE PERSON TO
2. CHARACTER Strike: NAME IS A 4 CHARACTER LONG, RIGHT JUSTIFIED, IF LESS THAN FOUR
3. AN INTEGER OF NINE OR LESS DIGITS, FOR USE IN GENERATING RANDOM NUMBERS."
)
6001 WRITE (6,3901)
3901 FORMAT (1x,' YOUR NAME ?')
READ (5,3904,END=6000) NAME
WRITE (6,3992)
3992 FORMAT (1x,'INRAND = ')
READ (5,3993) INRAND
3993 FORMAT (19)

C
SN=0
SS=0
WN=0
WS=0
THE PROGRAM NOW DETERMINES WHO MAKES THE FIRST MOVE. SELECTION IS BY A RANDOM PROCESS.

TURN=1 INDICATES THAT THE PLAYER MAKES THE FIRST PROPOSAL.
TURN=-1 INDICATES THAT THE COMPUTER MAKES THE FIRST PROPOSAL.

I=HEADS
CALL RANDOM
IF (YRAN.D.GE.0.5) I=TAILS
WRITE (6,203)
203 FORMAT (I5) A TOSSED COIN WILL BE USED TO DETERMINE WHO GOES FIRST. YOU MAY CALL "HEAD" OR "TAIL"!
READ (5,204) J
204 FORMAT (A4)
   IF (J.NE.I) GO TO 250
C
205 WRITE (6,205)
   FORMAT (//,'Congratulations, you won the toss of the coin. You
1may make the first proposal (concession point).')
   GO TO 200
C
201 WRITE (6,207) NC
202 FORMAT (//,'It is now move number',I3,',',I3,'. It is your turn to go
1first.')
200 TURN=1
   IF (NC.EQ.1) GO TO 206
   DO 207 NC = 0,1
   IF (RTABN(I,J).NE.COMPN(I,J)) GO TO 206
207 CONTINUE
   GO TO 208
206 WRITE (6,2000) (RTABS(I,J),I=1,3)
2000 FORMAT (///,'The approximate payoff matrix for the existing condi-
tions of strength and wealth is shown below. The indicated pay-
offs are to south (you).',//,'T28','South',//,'16','Strategy',10X,'11',5X,
3*X,'19',//,'0X','19X',***,10X,'0X','7X','1N',2X,
4*X,'7X',**,14,'216',//,'7X','1X',**,'7X','18',3X,'2',7X,**,'14',216,
5*X,'7X',**,11X,**,'7X','11',3X,'3',7X,**,'14',216,///)
   GO TO 210
208 WRITE (6,2009)
2009 FORMAT (///,'The approximate payoff matrix is unchanged from the last
move. You may use it as an aid in deciding your strategy
2 and cp.');//)
210 IF (NC.GT.2) GO TO 202
   WRITE (6,2001)
2001 FORMAT (//,'What is your proposal, strategy 1 (arm), 2 (maintain the
status quo), or 3 (disarm)?')
   GO TO 205
202 WRITE (6,2003)
2003 FORMAT (//,'What is your proposal, (1, 2, or 3) ?')
C
205 READ (5,206) CPS
206 FORMAT (I1)
   IF (CPS.NE.1.AND.CPS.NE.2.AND.CPS.NE.3) GO TO 410
C
   (Since CPS has been declared, the system may decide its
   strategy and CPN now. Without further reconsideration.)
   CALL STRATC(MOV,INCNT,PCNT,PCNTM,INCNTR,PCNTN,PCNTNM)
   CALL CPNDCN(MOV,INCNT,PCNTN,PCNTNM,INCNTR)
C
   IF (NC.GT.2) GO TO 2061
WRITE (6,207) CPN
GO TO 2065
2061 WRITE (6,2062) CPN
2062 FORMAT (//,' MY CP IS:','I2','')
C (DETERMINE IF SOUTH WANTS TO RENEGOTIATE:)
2065 IF (CPN.GT.1) GO TO 300
WRITE (6,2071)
2071 FORMAT (//,' NOTE THAT MY CP IS ARM.',//,' PLEASE INDICATE WITH 1A "YES" OR "NO" (RIGHT JUSTIFIED IN A3 FORMAT) IF YOU WISH TO RENE
2072 FORMAT (A3)
2074 I
2075 IF (I.EQ.NO) GO TO 300
2076 READ (5,2072)
2077 GO TO 300
C
250 WRITE (6,251)
251 FORMAT (//,' SORRY, YOU LOST THE TOSs OF THE COIN, THEREFORE I WILL
252 WRITE (6,252)
253 FORMAT (//,' IT IS NOW MY TURN TO GO FIRST.'))
254 TUPN=1
C
THE PROGRAM NOW DECIDES UPON A STRATEGY. BUT THE STRATEGY MAY BE
255 WRITE (6,256) NC
256 FORMAT (//,' IT IS NOW YOUR TURN TO MOVE NUMBER',I3,'. IT IS MY TURN TO GO FIRST.'))
257 TUPN-1
C
258 THE PROGRAM NOW DECIDES UPON A STRATEGY. BUT THE STRATEGY MAY BE
C
259 CHANGED IN LIGHT OF SOUTH'S CONCESSION BEFORE IT IS DECLARED.
C
260 CALL STRTGY(MOV,INCNT,PCNTM,INCNT,PCNTN,PCNTNM)
C
261 HAVING DECIDED UPON A TENTATIVE STRATEGY, THE SYSTEM NOW
C
262 CALL CPDCN(MOV,INCNT,PCNTM,INCNT,PCNTN,PCNTNM)
C
263 IF (NC.GT.2) GO TO 2075
WRITE (6,207) CPN
264 FORMAT (//,' MY PROPOSAL (CONCESSION POINT) WHERE 1=ARM, 2=MAINTAIN
265 WRITE (6,2076) CPN
266 FORMAT (//,' MY CP IS:','I2','')
C (DETERMINE IF SOUTH WANTS TO RENEGOTIATE:)
268 FORMAT (//,' MY CP IS:','I2','')
269 IF (CPN.GT.1) GO TO 2089
WRITE (6,2071)
4 READ (5,2072) I
    IF (I.EQ.NO) GO TO 2089
    WHOCAL=1
    CALL RENE(WPROBL,WHOCAL)
    (NOW ASK FOR SOUTH'S CONCESSION POINT.)
    C 2089 IF (NC,EQ.1) GO TO 208
        GO 2081 I=1,3
        IF (RTANS(I,J).NE.COMPN(I,J)) GO TO 208
    CONTINUE
    WRITE (6,2009)
    GO TO 2084
    2084 IF (NC,GT.2) GO TO 2085
    WRITE (6,2001)
    GO TO 209
    2085 WRITE (6,2003)
    209 READ (5,206) CPS
        IF (CPS.NE.1.AND.CPS.NE.2.AND.CPS.NE.3) GC TO 420
        THE PROGRAM NOW RECONSIDERS ITS PREVIOUSLY CHOOSEN STRATEGY SINCE
        SOUTH HAS NOW DECLARED HIS CPS.
        NC=NC+1
        CALL STRTY(MOV,INCNT,PCNT,PCNTM,INCTR,PCNTN,PCNTNM)
        NC=NC-1
        C 300 IF (NC,GT.2) GO TO 3001
        WRITE (6,3001)
        301 FORMAT (//,'IT IS NOW TIME TO CARRY OUT THE STRATEGY OF EACH SIDE
        THE COMPUTER'S MOVE IS LOCKED INTO THE SYSTEM.'//,'PLEASE INP
        320 WRITE (6,3001)
        3002 FORMAT (//,'WHAT IS YOUR STRATEGY ?'
        3003 READ (5,206) STRATS
        3004 WRITE (6,3002) STRATN, STRATS
        302 FORMAT (//,'THE ACTUAL STRATEGIES CHOSEN BY EACH OPPONENT ARE AS F
        303 FOLLOWS:'//,'5X,'NORTH(PLAYER):',T,5X,'SOUTH(PLAYER):',T)
        304 IF (STRATN.EQ.1.AND.STRATS.EQ.1) GO TO 3021
        IF (STRATN.EQ.1.AND.STRATS.EQ.3) GO TO 3022
        IF (STRATN.EQ.3.AND.STRATS.EQ.3) GO TO 3023
        IF (STRATN.EQ.3.AND.STRATS.EQ.1) GO TO 3024
        GO TO 302
        3021 FACTN=-1
        FACTS=-1
GO TO 303
3022 FACTXN=4
GO TO 303
3023 FACTXN=2
FACTXS=2
GO TO 303
3024 FACTXS=4
303 CALL PATUPD
C

317 PNOLED=PN
   (THIS VALUE IS LATER USED TO DETERMINE ECONOMIC CONDITIONS.)

   DETERMINE ISN
   IF (SN) 101,102,103
   SN NEGATIVE
   101 IF (SN.LT.-4) GO TO 1011
   IF (SN.LE.-2) GO TO 1012
   ISN=4
   GO TO 104
   1012 ISN=3
   GO TO 104
   1011 IF (SN.LT.-6) GO TO 1013
   ISN=2
   GO TO 104
   1013 ISN=1
   GO TO 104
C
   SN ZERO
   102 ISN=4
   GO TO 104
C
   SN POSITIVE
   103 IF (SN.GT.-4) GO TO 1031
   IF (SN.GE.-2) GO TO 1032
   ISN=4
   GO TO 104
   1032 ISN=5
   GO TO 104
   1031 IF (SN.GT.-6) GO TO 1033
   ISN=6
   GO TO 104
   1033 ISN=7
C
DETERMINE ISS

104 IF (SS) 111,112,113
111 IF (SS.LT.-4) GO TO 1111
   IF (SS.LT.-2) GO TO 1112
   ISS = 4
   GO TO 105
1112 ISS = 3
   GO TO 105
1111 IF (SS.LT.-6) GO TO 1113
1113 ISS = 1
   GO TO 105

SS ZERO

112 ISS = 4
   GO TO 105

SS POSITIVE

113 IF (SS.GT.-4) GO TO 1131
1131 IF (SS.GE.-2) GO TO 1132
   ISS = 4
   GO TO 105
1132 ISS = 5
   GO TO 105
1131 IF (SS.GT.-6) GO TO 1133
1133 ISS = 6
   GO TO 105

105 WN = WN + VALUE(ISN, STRATN) + ECNXR
   WS = WS + VALUE(IS, STRATS) + ECNXS
   JSTN = STRATN - 2
   JSTS = STRATS - 2
   IF (JSTN) 106, 107, 108
106 SN = SN + 1
   GO TO 107
107 SN = SN + 1
108 SN = SN + 1
109 SS = SS + 1
   GO TO 120
110 SS = SS - 1
120 PN = PN + 2*VALUE(ISN, STRATN) + ECNXR - 5*(STRATN - 2) + FACTXR
   PS = PS + 2*VALUE(IS, STRATS) + ECNXS - 5*(STRATS - 2) + FACTXS
   RN = WN - WS
RS=SN-SS

(CALL ANALYSES PORTION OF STRTGY TO ANALYZE THE RESULTS OF THE
LAST MOVE TO DETERMINE IF STRTGY COULD HAVE DONE BETTER.)
NC=NC+1
CALL STRTGY(MOV, INCNT, PCNT, PCNTM, INCNTR, PCNTN, PCNTNM)
NC=NC-1

C
WHOCAL=0
NFLAG=0

(RENEGOTIATE FLAGS CLEARED SINCE MOVE IS COMPLETE.)

CALL ECONMY

(DETERMINE ECONOMIC CONDITIONS BY CALLING SUBROUTINE ECONMY.)
DO 121 I=1,3
DO 121 J=1,3
121 COMP(I,J)=RTABN(I,J)
CALL TABCHG
FACTX=0
FACTX$=0
NC=NC+1
CALL ECALC(ENACT, ESACT)
NC=NC-1
ENAPRTY(NC)=ENACT
ESAPRTY(NC)=ESACT
ENAVG=ENAVG+ENACT
ESAVG=ESAVG+ESACT

IS IT END OF GAME YET?

THE NUMBER OF MOVES IN THE GAME VARIES RANDOMLY BETWEEN 10 AND 50.

IF (NC.LE.10) GO TO 480
IF (NC.GE.50) GO TO 496
CALL RANDOM
IF (YRAN.D,GE.0.9) GO TO 490
IF (NC.GT.40 .AND. YRAN.D,GE.0.4) GO TO 490
IF (NC.GT.30 .AND. YRAN.D,GE.0.5) GO TO 490
IF (NC.GT.20 .AND. YRAN.D,GE.0.6) GO TO 490
480 NC=NC+1
WRITE (6,4801) WN,SN,PN,WS,SS,PS
4801 FORMAT (/,, 'THE RESULTS OF THE LAST MOVE ARE AS FOLLOWS: ', //,
12X,'WEALTH',5X,'STRENGTH',5X,'POINTS',5X,
2,'COMPUTER (NORTH) ',1X,'IP',4X,'IP',3X,'**',//,
3,'PLAYER (SOUTH) ',3X,'**',17,3X,'**',18,4X,'**',18,3X,'**',//)
IF (ABS(RS).GE.10) GO TO 492
IF (IABS(RS).GE.30) GO TO 493
IF (TURN) 201,201,255
C 492 IF (RS) 494,494,495
C 493 IF(RM) 496,496,497
C 410 WRITE (6,411)
411 FORMAT (/,1** YOUR SELECTION OF A CONCESSION POINT WAS NOT CORRECT
1** THE CP MUST BE A SINGLE INTEGER 1, 2, OR 3.**) PLEASE INDICATE
2** YOUR CP AS 1(ARM), 2(STATUS QUO), OR 3(DISARM):**
GO TO 205
420 WRITE (6,411)
GO TO 209
430 WRITE (6,431)
431 FORMAT (/,1** YOUR SELECTION OF A STRATEGY MUST BE A SINGLE INTEGER
1** 1, 2, OR 3.**) PLEASE INDICATE YOUR STRATEGY AS 1(ARM), 2(STATUS QUO), OR 3(DISARM):**
READ (6,206) STRATS
GO TO 3011
C 494 WRITE(6,4941)
4941 FORMAT (/,1** CONGRATULATIONS, YOU WON THIS GAME BASED ON YOUR
1** STRENGTH SUPERIORITY.**)
WON=1
NC=NC-1
GO TO 491
495 WRITE (6,4951)
4951 FORMAT (/,1** SORRY, YOU LOST THIS GAME BASED ON YOUR STRENGTH
1** DEFICIENCY.**)
WON=1
NC=NC-1
GO TO 491
496 WRITE (6,4961)
4961 FORMAT (/,1** CONGRATULATIONS, YOU WON THIS GAME BASED ON YOUR
1** WEALTH SUPERIORITY.**)
WON=1
NC=NC-1
GO TO 491
497 WRITE (6,4971)
4971 FORMAT (/,1** SORRY, YOU LOST THIS GAME BASED ON YOUR WEALTH DEFICIENCY.**)
WON=1
NC=NC-1
GO TO 491
498 WRITE (6,4981)
4991 FORMAT (/,1** THE GAME HAS NOW ENDED. THE PARTICIPANT WITH THE
1** MOST POINTS (POINTS = (2*WEALTH)+(5*STRENGTH) IS DECLARED THE WINNERS.**)
IF (PN-PS) 391,392,393
391 WON=1
GO TO 491
392  W0N=0
GO TO 491
393  W0N=1
GO TO 491
490  WRITE (6,4901) NC
4901  FORMAT (/13, ** A RANDOM SELECTION, THE UMPIRE HAS DECIDED TO **
       END THIS GAME AFTER **13** MOVES ** */, ** THE PARTICIPANT WITH THE MOST **
       2T POINTS (POINTS = 2 WEALTH * (STRENGTH)) IS DECLARED THE WINNER **
       */
       IF (PN>PS) 3911,3922,3933
3911  W0N=-1
GO TO 491
3922  W0N=0
GO TO 491
3933  W0N=1
491  WRITE (6,4911) WN,SN,PN,WS,SS,PS
4911  FORMAT (/2, ** THE FINAL GRAND TOTALS FOR THIS GAME ARE: **
       12X, ** WEALTH:5X,** STRENGTH:5X,** POINTS:5X,** 1X,** 10X,** 1X,** 12X,**
       27** 11X,** /
       IF (W0N) 394,395,396
394  WRITE (6,3941) PLAYER
3941  FORMAT (/2, ** X: A4, ** IS THE WINNER. THANK YOU FOR PLAYING. **
       GO TO 1000
395  WRITE (6,3951) PLAYER
3951  FORMAT (/2, ** THE GAME ENDED IN A TIE. THANK YOU FOR PLAYING. **
       GO TO 1000
396  WRITE (6,3961) PLAYER
3961  FORMAT (/2, ** THE COMPUTER WON THE GAME. THANK YOU FOR PLAYING. **
       C (UPDATE HISTORY:)
1000  APN=APN+PN
APS=APS+PS
IF (W0N) 161,161,160
C (IF THE COMPUTER WON THE GAME, INSERT THE STRATEGY PARAMETERS
C INTO THE HISTORY OF THE PLAYER:)
160  CHAR=RLINK3(CATLG)
PARAM=LLINK4(CHAR)
NAME4(PARAM)=P(1)
RLINK4(PARAM)=P(2)
LLINK4(PARAM)=P(3)
DOWN4(PARAM)=P(4)
GO TO 162
161  CHAR=RLINK3(CATLG)
PARAM=LLINK4(CHAR)
NAME4(PARAM)=10
RLINK4(PARAM)=10
LLINK4(PARAM)=10
DOWN4(PARAM)=20
162 WRITE (6,1605) (P(I),I=1,4)
1605 FORMAT (/,'PARAMETERS AT END OF GAME A=E: ',Z15)
C
CHAR=RLINK2(CATLG)
TOTAL1=NAME4(LLINK4(PATRNS))
TOTAL2=RLINK4(LLINK4(PATRNS))
TOTAL3=LLINK4(LLINK4(PATRNS))
IF (TOTAL1.GT.TOTAL2.AND.TOTAL3.GT.TOTAL2) GO TO 150
IF (TOTAL1.GT.TOTAL2.AND.TOTAL3.GT.TOTAL3) GO TO 151
NAME2(RLINK4(CHAP))=2
GO TO 152
150 NAME2(RLINK4(CHAP))=1
GO TO 152
151 NAME2(RLINK4(CHAP))=3
152 ENAVG=ENAVG/NC
ESAVG=ESAVG/NC
NAME2(NAME4(CHAP))=ENAVG
DOWN2(NAME4(CHAP))=ESAVG
J=0
K=0
DO 157 I=1,NC
IF (ENAVR(I).EQ.10.OR.ESAVR(I).EQ.20) J=J+1
IF (ENAVR(I).EQ.30.OR.ESAVR(I).EQ.40) K=K+1
157 CONTINUE
IF (J.GT.K) HCLAS=2
IF (J.LT.K) HCLAS=3
IF (J.EQ.K) HCLAS=1
DOWN2(RLINK4(CHAP))=HCLAS
CALL TYPELBI(INCNT,WON,INCNT)
CALL SELDB(INCNT,WON)
CALL DPATNS
GO TO 6001
6000 WRITE (6,702)
702 FORMAT(/,'DO YOU WANT TO PRINT OUT THE PERMANENT MEMORY. YES OR NO
1. USE A3 FORMAT RIGHT JUSTIFIED?/
READ(5,703) DU
703 FORMAT(A3)
CALL SAVE(INCNT,INCNT,DU)
C
C
STOP
END
SUBROUTINE TABCHG

THIS SUBROUTINE UPDATES THE TABLES AS A RESULT OF THE LAST MOVE. IT ALSO DETERMINES IF AN "ECONOMIC CONDITIONS" CHANGE IS DUE. ECONOMIC CHANGES ARE RANDOMLY GENERATED AT RANDOM TIMES.

IMPLICIT INTEGER (A-W)
COMMON/JEC,VALUE(7,3),RTARS(2,3),ATARM(3),ATARS(3),
ISN,SS,WN,WS,PS,RW,MC,INRAND,YRAND,EFLAG,CPN,CPS,STRATN,STRATS,
2PNOLD,1SN,ISS,ECNXN,ECNXS,PN,PS,FACTXN,FACTXS

C

1 TI=1
   TJ=1
   TTN=2*VALUE(ISN,TI)-5*(TI-1)
   TTS=2*VALUE(ISS,TJ)-5*(TJ-1)
   RTARS(TI,TJ)=TTS-TTN
   IF(TJ.EQ.3) GO TO 3
   TJ=TJ+1
   GO TO 2

2 ATABN(TI)=TTN
   ATABS(TI)=2*VALUE(ISS,TI)-5*(TI-2)
   IF(TI.EQ.3) GO TO 10
   TI=TI+1
   TJ=1
   GO TO 1

3 PTARN(1,1)=RTABN(1,1)-1
   RTARS(1,1)=RTARS(1,1)-1
   RTARB(1,3)=RTABN(1,3)+4
   RTABS(3,1)=RTABS(3,1)+4
   RTAR(3,3)=RTABN(3,3)+2
   RTAB(3,3)=RTABS(3,3)+2
   ATARN(1)=ATARN(1)+2
   ATARS(1)=ATARS(1)+2
   ATABN(3)=ATABN(3)+1
   ATABS(3)=ATABS(3)+1
   (THE ATABN(6) FACTORS APPROXIMATE THE GAINS FOR COOPERATING IN DISARMING, OR FOR ONE SIDE ARMING WHILE THE OTHER DISARMS.)

4 THIS Completes THE TABLE CHANGES. THE ECONOMIC CONDITION IS NOW DETERMINED. THE FIRST STEP IS TO DETERMINE IF A CHANGE IS DUE. IF IT IS, THE CONDITIONS FOR THE OPPONENTS (COMPUTER IS "NORTH", OPPONENT IS "SOUTH") ARE PLACED INTO PARAMETERS ECNXN AND ECNXS RESPECTIVELY.

5 CALL RANDOM
   IF(YRAND.GE.0.6) GO TO 30
   (STMT 30 IS "RETURN", BRANCH THERE IF NO CHANGE IS DUE)
IN ORDER TO DETERMINE THE ECONOMIC CONDITIONS, A RANDOM NUMBER MUST FIRST BE GENERATED.

CALL RANDOM

THE ECONOMIC CONDITION FOR THE COMPUTER (NORTH) IS CALCULATED FIRST AND PLACED INTO ECNXN.

YRAND IS THE RANDOM NUMBER USED TO DETERMINE THE ECONOMIC CHANGE.

IF (YRAND.GE.0.7) GO TO 35
IF (YRAND.LE.0.3) GO TO 33
ECNXN=0
GO TO 40
33 IF (YRAND.LE.0.1) GO TO 34
ECNXN=-1
GO TO 40
34 ECNXN=-2
GO TO 40
35 IF (YRAND.GE.0.9) GO TO 36
ECNXN=1
GO TO 40
36 ECNXN=2

THE ECONOMIC CONDITION FOR SOUTH IS THEN CALCULATED AND PLACED INTO ECNXS. IT USES A DIFFERENT RANDOM NO. THAN FOR NORTH.

40 CALL RAND
IF (YRAND.GE.0.7) GO TO 45
IF (YRAND.LE.0.31) GO TO 43
ECNXS=0
RETURN
43 IF (YRAND.LE.0.1) GO TO 44
ECNXS=-1
RETURN
44 ECNXS=-2
RETURN
45 IF (YRAND.GE.0.9) GO TO 46
ECNXS=1
RETURN
46 ECNXS=2
RETURN
END
SUBROUTINE RANDOM

THE PURPOSE OF THIS SUBROUTINE IS TO GENERATE A FLOATING POINT
RANDOM NUMBER IN THE RANGE 0-1.0. THE PROCEDURE IS ESSENTIALLY
IDENTICAL TO THE "SCIENTIFIC SUBROUTINE PACKAGE" PROEDURE "RANUN".

PASSING PARAMETER INRAND IS USED TO START THE PROCESS OF RANDOM
NUMBER GENERATION. IT IS ANY ODD INTEGER NUMBER WITH NINE OR LESS
DIGITS INPUT IN A (HOPEFULLY) RANDOM MANNER WHEN THE PLAYER
INITIALLY SIGNS INTO THE SYSTEM. AFTER THE FIRST ENTRY TO THIS
SUBROUTINE IN ANY ONE GAME, INRAND BECOMES THE PREVIOUS VALUE OF
INRAND COMPUTED BY THIS SUBROUTINE.

IMPLICIT INTEGER (A-W)
COMMON/JEC/VALUE(7,3),RTABN(3,3),RATABS(3,3),ATABN(3),ATARS(3),
SN,SS,WN,WS,RN,RK,INRAND,YRAND,EFLAG,CPN,CP,S,TPATHN,STRATS,
FNO(1,3),ISN,ECNXS,PN,PS,FACTNX,FACTXS
INRAND=INRAND*65536
IF (INRAND) 31,32,32
31 INRAND=INRAND+2147483647+1
32 YRAND=INRAND*.46566139-9
RETURN

SUBROUTINE INITL

THIS SUBROUTINE INITIALIZES AVAILABLE CELLS FOR 250,333,
AND 500 NODES

IMPLICIT INTEGER (A-W)
COMMON/T,AVAIL4,AVAIL3,AVAIL2,NAME4(1000),NAME2(1000),NAME3(1000)
COMMON/TN,TDR,RTM,CTRL,CHAR,MIST,PLAYER,CLLEN,PATRNS,PATRXN
DIMENSION RLINK4(1000),LLNK4(1000),DOWN4(1000),RLINK3(1000),DOWN3
1(1000),DOWN2(1000)
EQUIVALENCE (NAME4(4),RLINK4(3),LLNK4(2),DOWN4(1)),(NAME2(2),DOWN
12(1)),(NAME3(3),RLINK3(2),DOWN3(1))
N=569
M=999
DO 10 I=4,N,4
10 NAME4(I)=I+1
AVAIL4=1
NAME4(1000)=0
DO 11 I=2,M,2
11 NAME2(I)=I+1
NAME2(1000)=0
AVAIL2=1
DO 12 I=3,P,3
   NAME3(I)=I+1
12 AVAIL3=1
   NAME3(999)=0
RETURN
END

SUBROUTINE GET2(K)
   THIS SUBROUTINE GEES A CELL FROM THE LIST OF AVAILABLE
   CELLS IN AVAIL2. "K" IS THE PASSING PARAMETER FOR SUBSCRIP
   TION OF CELL OBTAINED. "AVAIL2" IS THE POINTER TO THE NEXT
   CELL IN THE LIST OF AVAIL2 CELLS.
   IMPLICIT INTEGER (A-W)
   COMMON/T/AVAIL4,AVAIL3,AVAIL2,NAME4(1000),NAME2(1000),NAME3(1000)
   COMMON/TO/TOP,BOTM,CATLG,CHAR,HIST,PLAYER,CATLGN,PATRNS,PATRNN
   DIMENSION RLINK4(1000),LLINK4(1000),DOWN4(1000),PLINK3(1000),DOWN3
   12(11),NAME3(3),PLINK3(2),DOWN3(11)
   IF(AVAIL2.EQ.0) GOTO 2
   K=AVAIL2
   AVAIL2=DOWN2(AVAIL2)
   RETURN
2 WRITE(6,3)
   FORMAT(1HO,5X,'UNDERFLOW EXISTS IN AVAIL2 LIST OF AVAILABLE CELL
   IS.')
   RETURN
END

SUBROUTINE GET3(L)
   THIS SUBROUTINE GEES A CELL FROM THE LIST OF AVAILABLE
   CELLS IN AVAIL3. "L" IS THE PASSING PARAMETER FOR SUBSCRIP
   TION OF CELL OBTAINED. "AVAIL3" IS THE POINTER TO THE NEXT
   CELL IN THE LIST OF AVAIL3 CELLS.
   IMPLICIT INTEGER (A-W)
   COMMON/T/AVAIL4,AVAIL3,AVAIL2,NAME4(1000),NAME2(1000),NAME3(1000)
   COMMON/TO/TOP,BOTM,CATLG,CHAR,HIST,PLAYER,CATLGN,PATRNS,PATRNN
   DIMENSION RLINK4(1000),LLINK4(1000),DOWN4(1000),PLINK3(1000),DOWN3
   12(11),NAME3(3),PLINK3(2),DOWN3(11)
IF(AVAIL3.EQ.0) GOTO 2
L=AVAIL3
AVAIL3=DOWN3(AVAIL3)
RETURN
WRITE(6,3)
FORMAT(1HO,5X,'UNDERFLOW EXISTS IN AVAIL3 LIST OF AVAILABLE CELL
IS:')
RETURN
END

SUBROUTINE GET4(J)

THIS SUBROUTINE GETS A CELL FROM THE AVAILABLE CELLS IN
AVAIL4. "J" IS THE PASSING PARAMETER FOR SUBSCRIPT OF
CELL OBTAINED. "AVAIL4" IS THE POINTER TO NEXT CELL IN
THE LIST OF AVAIL4 CELLS.

IMPLICIT INTEGER (A-W)
COMMON/T/AVAIL4,AVAIL3,AVAIL2,NOME4(I000),NAME2(1000),NAME3(1000)
COMMON/TOP,BOTH,CATLG,CHAR,HIST,PLAYER,CATLGN,HITN,HITNN
DIMENSION RLINK4(1000),LLINK4(1000),DOWN4(1000),RLINK3(1000),DOWN3
1(1000),DOWN2(1000)
EQUIVALENCE (NAME4(4),RLINK4(3),LLINK4(2),DOWN4(1)),(NAME2(2),DOWN
12(1)),(NAME3(3),RLINK3(2),DOWN3(1))
IF(AVAIL4.EQ.0) GOTO 2
J=AVAIL4
AVAIL4=DOWN4(AVAIL4)
RETURN
WRITE(6,3)
FORMAT(1HO,5X,'UNDERFLOW EXISTS IN AVAIL4 LIST OF AVAILABLE CEL
IS:')
RETURN
END

SUBROUTINE COUNT(K1,J1,L1)

THIS SUBROUTINE COUNT THE NUMBER OF CELLS STILL AVAILABLE
"K1,J1,L1" ARE PASSING PARAMETERS THE PROVIDE THE COUNT
FOR EACH LIST. K1==AVAIL4,J1==AVAIL2,L1==AVAIL3.

IMPLICIT INTEGER (A-W)
COMMON/T/AVAIL4,AVAIL3,AVAIL2,NOME4(I000),NAME2(1000),NAME3(1000)
COMMON/TOP,BOTH,CATLG,CHAR,HIST,PLAYER,CATLGN,HITN,HITNN
DIMENSION RLINK4(1000),LLINK4(1000),DOWN4(1000),RLINK3(1000),DOWN3
1(1000),DOWN2(1000)
EQUIVALENCE (NAME4(4),RLINK4(3),LLINK4(2),DOWN4(1)),(NAME2(2),DOWN
12(1)\,,(\text{NAME3(3)},\text{RLINK3(2)},\text{DOWN3(1)})

K1=0
J1=0
L1=0
A4=\text{AVAIL4}
A3=\text{AVAIL3}
A2=\text{AVAIL2}

1 \text{IF}(A2 \text{EQ} 0) \text{GOTO 2}
A2=\text{DOWN2}(A2)
J1=J1+1
\text{GOTO 1}

2 \text{IF}(A4 \text{EQ} 0) \text{GOTO 3}
A4=\text{DOWN4}(A4)
K1=K1+1
\text{GOTO 2}

3 \text{IF}(A3 \text{EQ} 0) \text{GOTO 4}
A3=\text{DOWN3}(A3)
L1=L1+1
\text{GOTO 3}

4 \text{RETURN}
\text{END}

\text{SUBROUTINE STRTGY(MOV,INCNT,PCNT,PCNTM,INCNTP,PCNTN,PCNTNM)}

\text{************************************************************************}
\text{THE NINE SUBROUTINES WITHIN THIS SECTION SUPPORT THE}
\text{REASONING CAPABILITIES OF THE PROGRAM.}
\text{************************************************************************}

\text{THIS IS ONE OF THE KEY SUBROUTINES OF THE SYSTEM IN THAT IT IS}
\text{THE SUBROUTINE IN WHICH THE "THOUGHT PROCESSES" OCCUR. THIS}
\text{SUBROUTINE HAS THE FOLLOWING FUNCTIONS:}
1 \text{ANALYZE ALL FACTORS AVAILABLE AND DECIDE UPON A STRATEGY}
\text{FOR THE MOVE. THIS IS DONE BY USING A VOTING PROCESS ON A}
\text{POLYNOMIAL.}
2 \text{RECONSIDER THE STRATEGY CHOSEN IF THE COMPUTER WAS FIRST}
\text{TO DECLARE HIS CONCESSION POINT. THE RECONSIDERATION IS DONE IN}
\text{LIGHT OF THE PROPOSED STRATEGY OF SOUTH, THE COMPUTER'S OPPONENT.}
\text{(3) ANALYSIS OF THE COMPLETED MOVES TO DETERMINE IF THE}
\text{PARAMETERS USED FOR THE POLYNOMIAL COEFFICIENTS NEED BE TUNED UP}
\text{OR DOWN TO GIVE A BETTER SOLUTION.}

\text{IMPLICIT INTEGER (A-W)}
\text{COMMON/TDP4/SEQ(20),SCONTR}
COMMON/TOP3/TYPER(20),INIT(20)
COMMON/PAR/P(4),S(4),PRLS(5)
COMMON/JEC/JV(7,3),PTARN(3,3),RTABS(3,2),ATABN(3),ATABS(3),
1SN,SS,PN,WN,RS,RN,NC,INR,W,N,FRAN,EFLAG,CPN,PS,STRATN,STRAT,
2PNOLD,1SN,ISS,ECNXN,ECNXS,PN,PS,FACTXN,FACTXS
COMMON/EVIEW,E,TOPEN,TOPES
COMMON/RWP/RNARY(50),ESARY(50)
COMMON/ST/STRUE,LASTG0,PLEG,Turn,ANAL,RECON
COMMON/CLS/DEFE,DOE,OFFEN,OPEN,PMOV,PMOV
COMMON/TOP9/TYPER,TYPEI,TYPES,TYPOR
COMMON/TA/AVAIL4,AVAIL3,AVAIL2,NAME4(1000),NAME2(1000),NAME3(1000)
COMMON/TD/TOPM,RTNM,CATLG,CHAR,HIST,PLAYER,CATLGN,PATRNS,PATRNN
DIMENSION RLINK4(1000),LLINK4(1000),DOWN4(1000),RLNK3(1000),DOWN3
1(1000),DOWN2(1000)
EQUivalence (NAME4(4),RLINK4(3),LLINK4(2),DOWN4(1)),(NAME2(2),DOWN
12(1)),(NAME3(3),RLINK3(2),DOWN3(1))
DIMENSION CP(4)

(IF THE SYSTEM REMEMBERS HAVING PLAYED THIS OPPONENT, THE
PARAMETERS USED FOR THE COEFFICIENTS FOR THE TERM OF THE
POLYNOMIAL ARE TAKEN FROM THE HISTORY OF THIS PLAYER IN
LONG TERM MEMORY. IF NOT, THE PARAMETERS FOUND BELOW ARE
USED. -- THE FIRST TIME THIS SUBROUTINE IS CALLED IN A
GAME, THE PARAMETERS DESCRIBED ABOVE ARE USED. AFTER THE
FIRST MOVE, THESE PARAMETERS MAY BE MODIFIED BY THE ANALYSIS
PORTION OF THIS SUBROUTINE.)

IF (NC. GT. 1) GO TO 7
CHAR=RLINK3(4,CATLG)
IF (NAME4(LLINK4(CHAR)),EQ.0) GO TO 5000
P(1)=NAME4(LLINK4(CHAR))
P(2)=RLINK4(LLINK4(CHAR))
P(3)=LLINK4(LLINK4(CHAR))
P(4)=DOWN4(LLINK4(CHAR))
WRITE (6,61) (P(I),I=1,4)
61 FORMAT (///,4F11.4)
GO TO 7
6000 P(1)=10
P(2)=10
P(3)=10
P(4)=20
WRITE (6,61) (P(I),I=1,4)

THE FIRST STEP IS TO DETERMINE IF SOUTH'S CONCESSION POINT WAS
DECLARED FIRST. IF SO, THIS CPS IS USED IN DETERMINING THE
FIRST TERM OF THE POLYNOMIAL. IF NOT, THE FIRST TERM IS NOT
USED UNTIL THE TIME OF RECONSIDERATION OF THE STRATEGY AFTER
SOUTH HAS DECLARED HIS CPS.)

7 IF (TURN) 8, 8.90
8 IF (TURN) 9, 9.90
9 IF (RECON) 9, 9.90
10 SET THE RECON FLAG POSITIVE IN ANTICIPATION OF FUTURE RECON.
11 CF[1]=0
12 GO TO 20
13
14 IF (ANAL) 5, 1.70
15 (THE ANAL FLAG IS USED TO DETERMINE IF THE ANALYSIS PORTION OF
16 THE ROUTINE IS TO BE USED, BYPASSING THE STRATEGY PORTION.)
17
18 (THIS STEP OF THE STRATEGY PORTION OF THE SUBROUTINE IS TO
19 DETERMINE IF THE COMPUTER IS IN DANGER OF LOSING BECAUSE OF
20 RELATIVE STRENGTH OR RELATIVE WEALTH DEFICIENCIES. IF SO,
21 IMMEDIATE ACTION MUST BE TAKEN TO INCREASE STRENGTH OR WEALTH
22 ACCORDINGLY.)
23 IF (RS, GT, -7) GO TO 2
24 IF (TURN, GT, 0, AND, CPS, EQ, 1, AND, SS, GE, 4) GO TO 101
25 STRATN=1
26 GO TO 4
27 101 WRITE (6, 1010)
28 1010 FORMAT (/2, 'IT IS REQUESTED THAT WE RENEGOTIATE,'/
29 '1 YOU ARE QUITE STRONG, YET YOU PROPOSE ARMING EVEN FURTHER.'/
30 '2 AFTER RENEGOTIATING, WHAT IS YOUR NEW CP ?')
31 READ (9, 1011) CPS
32 1011 FORMAT (11)
33 STRATN=1
34 GO TO 4
35 IF (RS, GT, -23) GO TO 10
36 IF (ISN, EQ, 1) GO TO 3
37 STRATN=3
38 GO TO 4
39 STRATN=2
40 ANAL=-10
41 RETURN
42 ANAL=0
43 RECON=-10
44 RETURN
45
46 (IF SOUTH HAS DECLARED A CPS, THIS MUST BE ANALYZED AND A
10 ANAL=10
   (THIS SETS THE ANAL FLAG SO THAT ANALYSIS WILL
   BE PERFORMED WHEN REQUESTED.)
   GO TO (11,12,12), CPS
11 IF (SS.LE.4) GO TO 12
   (A BRANCH TO 11 INDICATES CPS WAS TO ARM.)
   (NO BRANCH INDICATES THAT THE COMPUTER'S OPPONENT IS QUITE
   STRONG ALREADY, YET HE IS ARMING EVEN FURTHER; CONSIDER
   RENEGOTIATION. BRANCHING TO 12 INDICATES NO RENEGOTIATE.)
   IF (NFLAG.EQ.1) GO TO 12
   (NFLAG=1 INDICATES THAT WE HAVE RENEGOTIATED, BUT SOUTH
   PERSISTS IN ARMING.)
   NFLAG=1
   IF (RS.LT.-2) GO TO 1111
   (NO BRANCH INDICATES THAT BOTH SIDES HAVE BEEN ARMING GREATLY.)
   PROBLM=1
   (I.E., "LET'S QUIT THE ARMS RACE.")
   CALL RENEG(PROBLM,WHOICAL)
   GO TO 10
1111 IF (NAME=E(LINKA(PATRNS))/NC)
   IF (Z.LE.0.5) GO TO 1112
   PROBLM=2
   (I.E., SOUTH IS QUITE STRONG, HAS MUCH STRENGTH SUPERIORITY,
   AND HAS ARMED OVER 50% OF THE TIME.)
   CALL RENEG(PROBLM,WHOICAL)
   GO TO 10
1112 J=0
   MOVES=DCWN4(PATRNS)
1113 IF (NAME=E(MOVES).EQ.1) J=J+1
   IF (DCWN2(MOVES).EQ.0) GO TO 1114
   MOVES=DCWN2(MOVES)
   GO TO 1113
1114 MOVES=DCWN4(PATRNS)
   IF (J.LT.2) GO TO 12
   J=0
   PROBLM=3
   (I.E., SOUTH IS QUITE STRONG AND HAS MUCH STRENGTH SUPERIORITY,
   YET HAS ARMED AT LEAST 2 OUT OF THE LAST 3 MOVES.)
   CALL RENEG(PROBLM,WHOICAL)
   GO TO 10
C
   (END CHECK FOR RENEGOTIATE BLOCK.)
(A BRANCH TO 12 INDICATES THAT CPS WAS TO MAINTAIN THE STATUS
QUO OR DISARMING. PROCEEDING HERE THROUGH THE LAST SECTION
INDICATES THAT CPS WAS TO ARM, BUT THE DECISION WAS MADE NOT
TO RENEGOTIATE.)

12 CALL ENPAT(PTRUTH)
   (PTRUTH=0 IMPLIES SOUTH IS PROBABLY NOT TELLING THE TRUTH,
   PTRUTH=1 IMPLIES HE PROBABLY IS.)
   IF (PTRUTH.EQ.0) GO TO 121
   PRALS(1)=CPS
   I=CPS
   GO TO 13

121 I=DOWN2(RLINK4(CHAR1))
   GO TO (122,123,124),I
122 PRALS(1)=I
   J=1
   GO TO 13
123 PRALS(1)=CPS
   J=CPS
   GO TO 13
124 PRALS(1)=3
   J=3

(NEXT CHECK WEIGHTED EN)

13 TOPEN=EN
   IF (NAME3(TOPEN).LT.25) GO TO 131
   (NO BRANCH INDICATES GOOD RELIABILITY)
   PRALS(2)=CPS
   GO TO 14
   (IF BRANCH TO 131, CHECK TO SEE IF EN IS INCREASING. IF SO,
   ASSUME HE IS TELLING TRUTH TO IMPROVE HIS RELIABILITY IMAGE.)
131 I=NAME3(DOWN3(TOPEN))
   IF (NAME3(TOPEN).GT.1) GO TO 132
   PRALS(2)=J
   GO TO 14
132 PRALS(2)=CPS

(NEXT CHECK AGGRESSIVENESS CLASS, I.E. HOW AGGRESSIVE HE IS.)

14 PRALS(3)=NAME2(RLINK4(CHAR1))
   (NOW CHECK HIS REST MOVES.)
   HISMAX=MAX0(ATAR5(1),ATAR5(2),ATAR5(3))
   DO 15 I=1,3
IF (HISMAX.EQ.ATABS(I)) GO TO 152
15 CONTINUE
WRITE (6,151)
151 FORMAT ('///,*** S/R STRTGY ERRCR #1***')
RETURN
152 PRBS(I,J)=1
HISMAX=MAX0(RTABS(1,1),RTABS(1,2),RTABS(1,3),RTABS(2,1),
RTABS(2,2),RTABS(2,3),RTABS(3,1),RTABS(3,2),RTABS(3,3))
DO 153 I=1,3
153 IF (RTABS(I,J).EQ.HISMAX) GO TO 155
CONTINUE
WRITE (6,154)
154 FORMAT ('///,*** S/R STRTGY ERRCR #2***')
RETURN
155 PRBS(I,J)=J
C  (NOW DETERMINE HIS PROBABLE STRATEGY BASED ON CPS.)
S(I)=0
S(J)=0
DO 16 I=1,5
IF (PRBS(I).EQ.1) S(I)=S(I)+1
CONTINUE
HISMAX=MAX0(S(1),S(2),S(3))
DO 161 I=1,3
IF (HISMAX.EQ.S(I)) GO TO 1611
CONTINUE
WRITE (6,163)
RETURN
1611 S(I)=0
S(2)=0
S(3)=0
HISMAX=I
C  (PICK COMPUTER'S VOTE FOR STRATEGY SELECTION BASED ON CPS TO
MAXIMIZE RELATIVE GAIN)
S(I)=MAX0(KTABN(I,HISMAX),KTAZN(2,HISMAX),KTAZN(3,HISMAX))
DO 162 I=1,3
IF (S(I).EQ.KTABN(I,HISMAX)) GO TO 164
CONTINUE
WRITE (6,163)
163 FORMAT ('///,*** S/R STRTGY ERRCR #3***')
RETURN
164 S(I)=I
CF(1)=P(I)
IF (HISMAX.EQ.3 .AND. S(1).EQ.1) CF(1)=P(1)+2

17 IF (RECON) 20,20,50
   (THIS COMPLETES THE RECONSIDERATION PHASE)

   THE SECOND TERM OF THE POLYNOMIAL IS BASED ON "GOAL". THIS IS
   BASED ON ANALYSIS OF THE STRENGTH, WEALTH, AND POINTS SO FAR, AND
   SELECTION OF THE GOAL OF THIS MOVE: OPTIMUM STRATEGY, MAXIMUM
   GAIN, OR MINIMUM LOSS. A COEFFICIENT CF2 IS THEN ASSIGNED TO THIS
   TERM OF THE POLYNOMIAL.

   20 PP=PN-PS
   IF (PP) 22,23,24

   (IF COMPUTER'S POINTS LESS THAN PLAYER'S:)

   22 IF (SN.LT.-4) GO TO 221
   242 CALL MAXSTR(MAXGN)
   S(2)=MAXGN
   GO TO 25
   221 S(2)=1
   GO TO 25

   (IF COMPUTER'S POINTS = PLAYER'S POINTS:)

   23 IF (SN.LT.-4) GO TO 221
   242 CALL OPTSTR(STRATC)
   S(2)=STRAC
   GO TO 25

   (IF COMPUTER'S POINTS GREATER THAN PLAYER'S:)

   24 IF (SN) 241,242,243
   241 IF (RS) 221,242,243
   243 CALL MINSTR(MINLOS)
   S(2)=MINLOS

   (DETERMINE CF2.)

   THE THIRD TERM OF THE STRATEGY POLYNOMIAL IS BASED ON
   "LOOK-AHEAD" AND "GUESS-OPPOSITES". LOOK-AHEAD IS DETERMINED BY PATTERN MATCHING THE STRATEGIES
   EMPLOYED BY THE COMPUTER'S OPPONENT AGAINST THE STRATEGY LIBRARY
   IN LONG-TERM MEMORY, THEN FOLLOWING THE BEST DEFENSE AS ALREADY
   DETERMINED THEREIN. A "GUESS-OPPOSITE" AGAINST THIS STRATEGY IS INFREQUENTLY
   EMPLOYED WHERE IT CAN PROVIDE A GOOD RELATIVE GAIN AND SERVE TO
   CONFUSE SOUTH. IT IS NOT USED VERY OFTEN, BECAUSE THE SYSTEM
   STRIVES TO MAINTAIN A GOOD IMAGE OF RELIABILITY.
C
IF (NC.GT.2) GO TO 31
IF (NAME4(HIST)) 302,30,302
C
(NAME4(HIST)=0 IF WE HAVE NOT PLAYED THIS OPPONENT BEFORE AND
C
THUS HAVE NO PRIOR KNOWLEDGE OF HIS PROBABLE STRATEGY.)
30 IF (LINK4(HIST)) 305,301,305
305 NORSO=0
CALL CLSTYP(NORSO)
IF (PMoves.EQ.0) GO TO 301
S(3)=MAX0(RTABN1,PMoves),RTABN2,PMoves),RTABN3,PMoves)
GO 3051 I=1,3
IF (S(3).EQ.RTABN1,PMoves)) GO TO 3055
3051 CONTINUE
WRITE (6,3052)
3052 FORMAT ('*** S/R STRTGY ERROR #3A ***')
CF(3)=C
GO TO 40
3055 S(3)=1
CF(3)=P(3)
GO TO 40
301 CALL RANDOM
IF (YRAND.GT.0.33) GO TO 3011
S(3)=1
GO TO 3059
3011 IF (YRAND.GT.0.67) GO TO 3012
S(3)=2
GO TO 3059
3012 S(3)=3
3059 CF(3)=P(3)*2
GO TO 40
C
C (IF THIS OPPONENT HAS PLAYED BEFORE, ASSUME HE WILL FOLLOW THE
SAME INITIAL STRATEGY, THE COMPUTER THEN PICKS THE BEST
DEFENSE AGAINST THIS INITIAL STRATEGY.)
302 MOVE1=NAME4(NAME4(HIST))
MOVE2=RLINK4(NAME4(HIST))
GO TO (203,304),NC
303 S(3)=MAX0(RTABN1,MOVE1),RTABN2,MOVE1),RTABN3,MOVE1)
GO 3031 I=1,3
IF (S(3).EQ.RTABN1,MOVE1)) GO TO 3033
3031 CONTINUE
WRITE (6,3032)
3032 FORMAT ('///,1 *** S/R STRTGY ERROR #4 ***')
RETURN
3032 S(3)=1
CF(3)=P(3)
GO TO 40
304 S(3)=MAX0(RTABN1,MOVE2),RTABN2,MOVE2),RTABN3,MOVE2))
DO 3041 I=1,3
IF (S(3).EQ.RTABN(I,MOVE2)) GO TO 3043
3041 CONTINUE
WRITE (6,3042)
3042 FORMAT (//,'*** S/R STRTGY ERROR #5 ***')
RETURN
3043 S(3)=I
CF(3)=P(3)
GO TO 40
31 IF (NC..GT.4) GO TO 32
CALL CLSMVSI(MOVI,INCNT,PCNT,PCNTM,INCNT)
IF (DEFES.GT.0) GO TO 311
IF (BDEF..GT.0) GO TO 311
CF(3)=0
GO TO 40
C
311 S(3)=MAXO(RTABN(1,PMOVES),RTABN(2,PMOVES),RTABN(3,PMOVES))
DO 3111 I=1,3
IF (S(I).EQ.RTABN(I,PMOVES)) GO TO 312
3111 CONTINUE
WRITE (6,3143)
3143 FORMAT (//,'*** S/R STRTGY ERROR #6 ***')
CF(3)=0
GO TO 40
C
312 CALL CLSMVNI(MOVI,INCNT,PCNT,PCNTM,INCNT)
IF (DEFNI.GT.0) GO TO 313
IF (BDEF..GT.0) GO TO 313
S(3)=I
CF(3)=P(3)/2
GO TO 25
C
(HAVE BOTH PMOVES AND PMOVEN:)
313 IF (I.EQ.PMOVEN) GO TO 314
S(3)=I
CF(3)=P(3)/2
GO TO 29
314 S(3)=PMOVEN
CF(3)=P(3)
GO TO 29
C
32 NORSO=C
CALL CLSTYP(NORSO)
IF (PMOVES..GT.0) GO TO 321
CF(3)=0
GO TO 40
321 S(3)=MAXO(RTABN(1,PMOVES),RTABN(2,PMOVES),RTABN(3,PMOVES))
DO 3211 I=1,3
IF (S(3).EQ. RTABN(I, PMOVES)) GO TO 322
CONTINUE
WRITE (6, 3243)
3243 FORMAT ("//, ""S/R STRATEGY ERROR \"***\""")
RETURN
C
322 NORSO = 1
CALL CLSTYP(NORSO)
IF (PMOVEN .GT. 0) GO TO 323
S(3) = I
CF(3) = P(3) / 2
GO TO 36
C
323 IF (I .EQ. PMOVEN) GO TO 324
S(3) = I
CF(3) = P(3) / 2
GO TO 38
324 S(3) = PMOVEN
CF(3) = P(3)
GO TO 39
C
C
C
C
C
NOW DETERMINE IF WE SHOULD GUESS OPPOSITE.)
38 I = NC-LASTGO
IF (I .LT. 5) GO TO 39
C (DONT GUESS OPPOSITE MORE THAN EVERY FIVE MOVES.)
C (ALSO DONT GUESS OPPOSITE IF COMPUTER'S RELIABILITY IS LOW.)
TOPE = ES
IF (NAME3(TOPES), LT, 40) GO TO 39
C (GUESS OPPOSITE, IF USED, IS AGAINST OUR PATTERN OF MOVES)
C IF (PMOVEN , EQ, 0) GO TO 39
STRUE = PMOVEN
LASTGO = NC
CF(3) = P(3) * 2
GO TO 40
C
C
C
C
C
C
C
C
C
NOW DETERMINE IF WE SHOULD GUESS OPPOSITE.)
38 I = NC-LASTGO
IF (I .LT. 5) GO TO 39
C (DONT GUESS OPPOSITE MORE THAN EVERY FIVE MOVES.)
C (ALSO DONT GUESS OPPOSITE IF COMPUTER'S RELIABILITY IS LOW.)
TOPE = ES
IF (NAME3(TOPES), LT, 40) GO TO 39
C (GUESS OPPOSITE, IF USED, IS AGAINST OUR PATTERN OF MOVES)
C IF (PMOVEN , EQ, 0) GO TO 39
STRUE = PMOVEN
LASTGO = NC
CF(3) = P(3) * 2
GO TO 40
C
39 CALL RANDOM
IF (YRAND .GT. 0.2) GO TO 40
IF (YRAND .GT. 0.1) GO TO 395
IF (S(3), EQ, 1) S(3) = 3
IF (S(3), EQ, 3) S(3) = 1
CF(3) = P(3) * 3
LASTGO = NC
GO TO 40
395  S(3)=2
CF(3)=P(3)*3
LASTGO=NC

THE FOURTH TERM OF THE POLYNOMIAL IS BASED ON THE EXPECTED
ECONOMIC CONDITIONS. COEFFICIENT CF4 IS ASSIGNED TO THIS TERM.

40 IF (EFLAG) 41, 42, 43

(EFLAG NEGATIVE INDICATES THAT A "BOOM" IS PROBABLY IN PROGRESS
AND THAT PRICES ARE HIGH, AND THIS IS A BAD TIME TO ARM.)

41 S(4)=3
CF(4)=P(4)
GO TO 50

(EFLAG ZERO INDICATES THAT THE PROBABLE STATE OF THE ECONOMY
IS "NORMAL", THUS NO SPECIAL CONSIDERATION FOR ECONOMIC
CONDITIONS NEED BE TAKEN INTO ACCOUNT.)

42 CF(4)=0
GO TO 50

(EFLAG POSITIVE INDICATES THAT A "RECESS" IS PROBABLY IN PROGRESS
AND THAT PRICES ARE LOW. IF STRENGTH IS NOT ESPECIALLY HIGH,
THIS IS A GOOD TIME TO ARM.)

43 IF (SN.6T.4) GO TO 431
S(4)=1
CF(4)=P(4)
GO TO 50

431 I=MAXO(ATABN(2),ATABN(3))
DO 432 I=2,3
IF (J.EQ.ATABN(1)) GO TO 434
432 CONTINUE
WRITE (6,433)
433 FORMAT (6,433)
*** S/R STRG/Y ERRCR #9 ***
434 S(4)=1
CF(4)=P(4)

NOW COMBINE ALL THE TERMS INTO THE POLYNOMIAL AND DETERMINE
STRAIN BY A VOTING PROCESS.)

50 PRBS(1)=0
PRBS(2)=0
PRBS(3)=0
DO 501 I=1,4
IF (S(I).EQ.1) PRBS(1)=PRBS(1)+CF(I)
IF (S(I).EQ.2) PRBS(2)=PRBS(2)+CF(I)
IF (S(I).EQ.3) PRBS(3)=PRBS(3)+CF(I)
501 CONTINUE
J=MAX0(PRBL$1),PRBL$2),PRBL$3))
DO 502 I=1,3
IF (J.EQ.PRBL$1)) GO TO 504
502 CONTINUE
WRITE (6,503)!!! S/R STRTGY ERRCR #11 !!!)
RETURN
504 IF (PRBL$1).EQ.PRBL$2).OR.PRBL$1).EQ.PRBL$3).OR.
1PRBL$2).EQ.PRBL$3)) GO TO 51
505 STRATN=1
RETURN
51 IF (PRBL$1).EQ.PRBL$2)) GO TO 511
IF (PRBL$1).EQ.PRBL$3)) GO TO 512
IF (J.NE.PRBL$2)) GO TO 505
STRATN=MAXO(AT=BN(1),AT=BN(3))
DO 510 K=2,2
IF (STRATN.EQ.AT=BN(K)) GO TO 5103
510 CONTINUE
5101 WRITE (6,5102)
5102 FORMAT (/'S/R STRTGY ERRCR #12 ** **)
RETURN
5103 STRATN=K
RETURN
511 IF (J.NE.PRBL$1)) GO TO 505
STRATN=MAXO(AT=BN(1),AT=BN(2))
DO 5110 K=1,2
IF (STRATN.EQ.AT=BN(K)) GO TO 5103
5110 CONTINUE
GO TO 5101
512 IF (J.NE.PRBL$1)) GO TO 505
STRATN=MAXO(AT=BN(1),AT=BN(3))
DO 5120 K=1,3,2
IF (STRATN.EQ.AT=BN(K)) GO TO 5103
5120 CONTINUE
GO TO 510;
(THE THIRD FUNCTION OF THIS SUBROUTINE IS TO ANALYZE THE
RESULTS OF THE LAST MOVE AND DETERMINE IF A CORRECT CHOICE
OF STRATEGY WAS MADE. IF SO, WELL AND GOOD, BUT IF NOT
THE PARAMETERS P(I) ARE CHANGED TO REFLECT WHAT MAY HAVE
BEEN A BETTER CHOICE. ANALYSIS IS NOT DONE IF THE "EMERGENCY"
PROCEDURES OF STATEMENT 1 OF THIS SUBROUTINE ARE CALLED OUT.)
C 7C ANAL=0
(THE RESFTS THE ANAL FLAG.)
RETURN
END

SUBROUTINE ECALC(ENACT,ESACT, 
IMPLICIT INTEGER (A-W)
COMMON/JEC/VALUE(7,3),RTABN(3,3),RTABS(3,3),ATABN(3),ATABS(3), 
1SN,SS,WN,V3,RS,RW,NC,ENR,RY,NR,RAND,RF,RFIL,AFIL,ON,CPS,STRTN,STRAT, 
2PNOLD,SN,IS,EN,E,ST,PEN,PEN,
COMMON/E,EE,ES,TOP,TOPES, 
COMMON/TC/AVAIL4,AVAIL3,AVAIL2,NAME4(1000),NAME2(1000),NAME2(1000) 
COMMON/T/CHAR,CHARHIST,PLAYER,CATLN,PATMS,PATNRN, 
DIMENSION RLINK4(1000),RLINK3(1000),DOWN4(1000),RLNK3(1000),DOWN3 
1(1000),DOWN2(1000)
EQUVALENCE (NAME4(4),RLINK4(3),LLINK4(2),DOWN4(1)),(NAME2(2),DOWN 
12(1)),(NAME3(3),RLINK3(2),DOWN3(1))

THE PURPOSE OF THIS SUBROUTINE IS TO DETERMINE THE WEIGHTED 
EN (NORTH'S ESTIMATE OF SOUTH'S RELIABILITY) AND THE WEIGHTED 
ES (NORTH'S ESTIMATE OF NORTH'S RELIABILITY).

KEY OF EACH MOVE'S EN/ES:
A. 50 == COMPLETELY TRUTHFUL  (CPS-STRAT= 0) 
B. 10 ==> AGGRESSIVELY NOT TRUTHFUL BY 2  (CPS-STRAT=+2) 
C. 20 ==> AGGRESSIVELY NOT TRUTHFUL BY 1  (CPS-STRAT=+1) 
D. 0  ==> PASSIVELY NOT TRUTHFUL BY 2  (CPS-STRAT=-2) 
E. 40 ==> PASSIVELY NOT TRUTHFUL BY 1  (CPS-STRAT=-1)

WEIGHTED EN OR ES IS DETERMINED FROM THE ABOVE STRAIGHT VALUES.
IT IS DETERMINED AS FOLLOWS:
1. WEIGHTED EN = (6*(THIS MOVE'S STRAIGHT EN) 
   + 3*(LAST WEIGHTED EN) 
   + 1*(NEXT TO LAST WEIGHTED EN))/10 

2. IF PLAYER WAS NOT TRUTHFUL, BUT THE WEIGHTED EN FOR 
   THIS MOVE IS GREATER THAN FOR THE LAST MOVE, 
   SUBTRACT 1 FOR A, 2 FOR B, OR 3 FOR C, D, OR E 
   (LETTER'S REFERS TO THE KEY ABOVE) 
   FROM THE PREVIOUS EN AND REFIGURE WEIGHTED 
   EN USING THIS VALUE AS THIS MOVE'S STRAIGHT EN IN 
   (1) ABOVE. (THIS IS TO PREVENT A DISHONEST CPS 
   FROM ACTUALLY INCREASING THE WEIGHTED EN BECAUSE 
   OF THE WEIGHTS ASSIGNED TO EACH MOVE'S VALUE.) 

{(THE FIRST STEP IS TO INITIALLY RUILL THE EN AND ES DEQUEUES.) 
IF (NC.GT.1) GO TO 1 } 

(FIRST, EN) 
CALL GET3(L)
DOWN3(L)=0
NAME3(L)=50
RO5EN=5
RLINK3(L)=1
CALL GET3(L)
DOWN3(L)=RO5EN
NAME3(L)=50
MIDEN=1
RLINK3(L)=2
CALL GET3(L)
DOWN3(L)=MIDEN
TOPEN=L
NAME3(L)=50
RLINK3(L)=6
EN=L
(THEN, ES:)
CALL GET3(L)
DOWN3(L)=0
POTES=L
NAME3(L)=50
RLINK3(L)=1
CALL GET3(L)
DOWN3(L)=POTES
NAME3(L)=50
MIDES=L
RLINK3(L)=3
CALL GET3(L)
DOWN3(L)=MIDES
TOPES=L
NAME3(L)=50
ES=L
RLINK3(L)=6
IF (NC.EQ.1) RETURN

NAME3(RO5EN)=NAME3(MIDEN)
NAME3(MIDEN)=NAME3(TOPEN)
NAME3(POTES)=NAME3(MIDES)
NAME3(MIDES)=NAME3(TOPES)
J=CPS-STRATS+2
K=CPN-STRATN+3
GO TO (2,3,4,5,6), J

(SOUTH WAS PASSIVELY DISHONEST BY 2:)
1 NAME3(TOPEN)=30
2 J=4
GO TO 10

(PASSIVELY DISHONEST BY 1:)
3 NAME3(TOPEN)=40
SUBROUTINE CPNDCN(MOV, INCNT, PCNTN, PCNTNM, INCNTR)

THIS SUBROUTINE DETERMINES THE CONCESSION POINT TO BE DECLARED
BY THE COMPUTER (NORTH), CPN. IT DOES THIS AFTER STRATN, THE
STRATEGY TO BE FOLLOWED, HAS BEEN DECIDED UPON BY S/R STRATG.

IMPLICIT INTEGER (A, W)
COMMON/ENP/enarry(50), esarry(50)
COMMON/JEC/VA: IF(7, 3), RTARNS(3, 3), RTRANS(3, 3), ATARNS(3), ATTRANS(3),
1 SN, SS, WN, WS, RN, RW, NC, INRND, YRAND, EFLAG, CPN, CPS, STRATN, STRATS,
2 PNO, PNO, PNO, PCN, PCNY, PN, PS, FACTXN, FACTXS
COMMON/TOP4/SEQ(20), SCONTR
COMMON/TOP3/TYPE(20), INIT(20)
COMMON/SEE/EN, ES, TOPEN, TOPE
COMMON/ST/TRUE, LASTGO, NFLAG, TURN, ANAL, RECON
COMMON/TOP/TOP, TYPE, TYSOU, TYNDR
COMMON/CLS/DEFES, DEFFS, OEFES, OEFFS, MOVEN, PMOVES
COMMON/TI/AVAL4, AVAL3, AVAL2, NAME4(1000), NAME2(1000), NAME3(1000)
COMMON/TD/TOP, RGM, CATLG, CHAR, HIS, PLAYER, CATLGN, PATRNS, PATRNN
DIMENSION RLINK4(1000), LLINK4(1000), ROWN4(1000), RLINK3(1000), DOWN3(11000), DOWN4(1000)
EQUIVALENCE (NAME4(4), RLINK4(3), LLINK4(2), DOWN4(1)), (NAME2(2), DOWN
12(1)), (NAME3(3), RLINK3(2), DOWN3(1))

(FIRST DECIDE IF NORTH'S RELIABILITY IS HIGH ENOUGH TO AVOID
A NON-TRUTHFUL DECLARATION. IF NOT, USE A RANDOM NUMBER TO
DECIDE IF A FALSE DECLARATION SHOULD BE MADE ANYWAY. THIS
IS TO PREVENT BULDO-UP OF A PATTERN OF UNRELIABILITY.

IF (NC.GT.2) GO TO 3
5 CALL RANDOM
6 IF (YRAND.GT.0.33) GO TO 1
7 CPN=STRATN
8 RETURN
1 IF (YRAND.GT.0.67) GO TO 2
9 CPN=2
10 RETURN
2 CPN=3
11 RETURN
3 IF (NC.GT.4) GO TO 10
4 IF (DEFES.GT.0.OR.BDEFS.GT.0) GO TO 4
4 CALL CLSMVN(MOV, INCNT, PCNTN, PCNTNM, INCNTR)
4 IF (PMOVEN.LE.0.0) GO TO 5
25  CPN=PMOVEN
RETURN
10  TOPES=ES
IF (NAME3(TOPES).GE.40) GO TO 20
TOPES=DOWN3(TOPES)
IF (NAME3(TOPES).LE.NAME3(TOPES)) GO TO 15
TOPEN=TOPES
TOPES=DOWN3(TOPES)
IF (NAME3(TOPEN).GT.NAME3(TOPES)) GO TO 20
( GO AHEAD IF RELIABILITY INCREASING. )
C
15  TOPEN=EN
TOPES=ES
CALL RANDOM
IF (YRAND.LT.0.3) GO TO 20
CPN=STDTN
RETURN
20  TOPEN=EN
TOPES=ES
IF (PMOVES.GT.0) GO TO 21
NORSD=1
CALL CLSYNP(NORSO)
21  IF (PMOVEN.GT.0) GO TO 25
CALL RANDOM
IF (YRAND.LT.0.5) GO TO 6
IF (STDTN.EQ.1) CPN=3
IF (STDTN.EQ.2) CPN=2
IF (STDTN.EQ.3) GO TO 5
RETURN
END

SUBROUTINE PENG(PROMW,WHOCAL)

THIS SUBROUTINE IS CALLED UPON IF EITHER NORTH OR SOUTH WISH TO
NEGOTIATE THE OTHER'S CONCESSION POINT CALLING FOR ARMING.

IMPLICIT INTEGER (A-H)
COMMON/ST/ST TRUE  LAST=0,NFLAG,TURN, ANAL, RECON
COMMON/JEC/VALUE(7,3),RTABN(3,3),RTARS(3,3),ATABN[3],ATARS[3],
ISN,SS,WN,WS,RS,RW,NC,INRANC,YRND,EN,PN,CPN,CPS,STDTN,STRTS,
2PNOLO,ISN,SS,ECNXN,ECNXS,PN,PS,FACTN,FACTXS
C
IF (WHOCAL.EQ.1) GO TO 100
(C A BRANCH INDICATES THAT THE PLAYERS ASKED FOR NEGOTIATION, )
( A NO BRANCH INDICATES THAT THE COMPUTER ASKED. )
GO TO (10,20,30),PROBLEM
10 WRITE (6,11)
11 FORMAT (/,I4,1X, 'OUR STRENGTHS AND MOVES SO FAR HAVE INDICATED THAT WE')
1 ARE CONDUCTING AN ARMS RACE. A RENEGOTIATION IS REQUESTED.
WRITE (6,12)
12 FORMAT ('I AFTER RENEGOTIATING, WHAT IS YOUR NEW CONCESSION PT?')
READ (5,13) CPS
13 FORMAT (11)
RETURN
20 WRITE (6,21)
21 FORMAT ('/// IT IS REQUESTED THAT WE RENEGOTIATE. YOU HAVE ARMED
OVER 50% OF THE MOVES, AND IF THIS keeps UP, WE WILL BOTH GO BROKE
/// CONDUCTING AN ARMS RACE.')
WRITE (6,12)
READ (5,13) CPS
RETURN
30 WRITE (6,31)
31 FORMAT ('/// IT IS REQUESTED THAT WE RENEGOTIATE. YOU ARE ALREADY
1 QUITE STRONG, YET YOU HAVE ARMED AT LEAST TWO OUT OF THE LAST',
2/' THREE MOVES.')
WRITE (6,12)
READ (5,13) CPS
RETURN
C 100 CALL MAXSTRP(MAXGN)
C (THE PROGRAM ASKS FOR THE MAXIMUM GAIN STRATEGY IN ORDER TO
C APPEASE THE PLAYER, YET BENEFIT AS MUCH AS POSSIBLE.)
C STRATN-MAXGN
C CALL RANDOM
IF (YRAND.GT.0.8.AND.STRATN.EQ.1) STRATN=2
CALL RANDOM
IF (YRAND.GT.0.67) GO TO 110
IF (YRAND.GT.0.33) GO TO 120
CPN=STRATN
GO TO 40
110 CPN=2
GO TO 40
120 CPN=3
GO TO 40
40 WRITE (6,101) CPN
101 FORMAT ('/// BASED ON YOUR REQUEST FOR RENEGOTIATION, I HAVE CONE
1 CERED MY CONCESSION POINT. MY NEW CP IS TO: ',12,
WELCOME=0
RETURN
END

SUBROUTINE OPTSTR(Strato)
C THIS SUBROUTINE DETERMINES THE OPTIMUM STRATEGY BASED ON ZERO-SUM
C TWO PERSON RECTANGULAR GAME THEORY. THE OPTIMUM STRATEGY IS
C RETURNED BY THE PARAMETER STRATO.
IMPLICIT INTEGER (A-W)
COMMONT JEC/VALUE(7,3),RTABN(3,3),RTABS(3,3),ATARN(3),ATARS(3),
1SN,SS,WN,WS,RS,WN,NC,INRAND,PRAND,EFLAG,CPN,CP,S,STRATN,STRAT5,
2PNOLD,ISN,ISS,ECNXX,ECNXS,PN,PS,FACTXN,FACTXS
DIMENSION Q(3),W(3)

THE SUBROUTINE FIRST DETERMINES IF A SADDLE POINT EXISTS.

DO 10 I=1,3
  Q(I) = MINQ(RTABN(I,1),RTABN(I,2),RTABN(I,3))
10  Q(I) = MAXQ(RTABN(I,1),RTABN(I,2),RTABN(I,3))

OMAX = MAXQ(Q(1),Q(2),Q(3))
WMIN = MINQ(W(1),W(2),W(3))
IF (OMAX .NE. WMIN) GO TO 30
DO 11 I=1,3
  IF (OMAX .EQ. Q(I)) GO TO 13
11 CONTINUE
WRITE (6,12)
12 FORMAT ('S/R OPTSTR ERROR #1 *****')
STOP

DO 13 J=1,3
  IF (WMIN .EQ. W(J)) GO TO 22
13 CONTINUE
WRITE (6,15)
15 FORMAT ('S/R OPTSTR ERROR #2 *****')

L = 1
16 IF (RTABN(L,J) .EQ. OMAX) GO TO 40
17 IF (L .EQ. 3) GO TO 19
K = L + 1
DO 18 L = K, 3
  IF (OMAX .EQ. Q(L)) GO TO 16
18 CONTINUE
19 J = J + 1
DO 20 L = J, 3
  IF (WMIN .EQ. W(L)) GO TO 21
20 CONTINUE
GO TO 30
21 J = L
GO TO 22
40 STRATOL = L
RETURN

IF NO SADDLE POINT EXISTS, THE CONCEPT OF DOMINENCE IS USED TO
ATTEMPT TO REDUCE THE SIZE OF THE MATRIX.
30 L=3  
   K=1
31 IF (RTABN(I,1)GE.RTABN(K,1).AND.RTABN(I,2)GE.RTABN(K,2).AND.  
   RTABN(I,3)GE.RTABN(K,3)) GO TO 35 
   GO TO 31 
35 IF (RTABN(I,1)GE.RTABN(L,1).AND.RTABN(I,2)GE.RTABN(L,2).AND.  
   RTABN(I,3)GE.RTABN(L,3)) GO TO 36 
36 STRATO=1 
   RETURN 

THE STMT 34 BRANCH INDICATES THAT NO OPTIMUM STRATEGY CAN BE  
EASILY DETERMINED. IN THIS CASE, THE S/R RETURNS A STRATEGY  
RECOMMENDATION BASED ON STRENGTH.

34 IF (SN.LE.-2) GO TO 39 
   STRATO=3 
   RETURN 
39 STRATO=1 
   RETURN 
END

SUBROUTINE MAXSTR(MAXGN)

THIS SUBROUTINE DETERMINES THE MAXIMUM GAIN STRATEGY FROM THE  
RELATIVE TABLE (RTABN) OF POINTS. THE SELECTED STRATEGY NUMBER  
IS RETURNED BY PARAMETER MAXGN.

IMPLICIT INTEGER (A-W) 
COMMON /JEC/VALUE(7,3),RTABN(3,3),RTAPS(3,3),ATABN(3),ATAPS(3),  
ISN,SS,SN,WS,SR,SN,INRAND,VRAND,EFLAG,CPN,CPS,STRATN,STRATS,  
2PVOID,LSS,ISN,SS,ECX,ECNX,SN,PS,FACXN,FACXS 
DIMENSION Q(3)

DO 12 I=1,3 
12 Q(I)=MAX0(RTABN(I,1),RTABN(I,2),RTABN(I,3)) 
   MAXGN=MAX0(Q(1),Q(2),Q(3)) 
DO 14 I=1,3 
14 IF (MAXGN.EQ.0(I)) GO TO 13 
CONTINUE
WRITE (6,1)"
FORMAT (//,' S/R MAXSTR ERROR ************')
RETURN
13 MAXEN=1
RETURN
END

SUBROUTINE MINSTR(MINLCS)
THIS SUBROUTINE DETERMINES THE STRATEGY FOR MINIMUM RISK OF LOSS
OF POINTS RELATIVE TO THE OPPONENT. THE SELECTED STRATEGY IS
RETURNED BY THE PARAMETER MINLCS.

IMPLICIT INTEGER (A-W)
COMMON /JEC/VALUE(7,3),RTABN(3,3),RTABR(3,3),ATABN(3),ATABR(3),
1SN,SS,WN,WS,RS,RW,NC,INRAND,VRANEF,CPN,CP,SP,SPAM,STRAT5,
2PNOLD,ISN,ISS,ECNXX,ECNXS,PN,PS,FACTXN,FACTXS
DIMENSION G(3)

DO 10 I=1,3
Q(I)= MIN[RTABN(I,1),RTABN(I,2),RTABN(I,3)]
OMAX=MAX[Q(1),Q(2),Q(3)]
DO 12 I=1,3
IF (OMAX.EQ.Q(I)) GO TO 13
CONTINUE
WRITE (6,1)
1 FORMAT (//,' S/R MINSTR ERROR ************')
RETURN
13 MINLOS=I
RETURN
END

SUBROUTINE ECONOMY

THIS SUBROUTINE DETERMINES IF THE STATE OF THE ECONOMY HAS CHANGED,
AND IF SO, ATTEMPTS TO DETERMINE THE ECONOMIC CONDITIONS AND SET A
FLAG (EFLAG) ACCORDINGLY.

THE ECONOMIC CONDITION IS RANDOMLY GENERATED AT RANDOM TIMES, BUT
THE AMOUNT OF CHANGE TO COSTS IS NOT FACTORED INTO THE TABLES, AND
HENCE IS UNKNOWN TO THE OPPONENTS (COMPUTER AND PLAYER).

IMPLICIT INTEGER (A-W)
COMMON /JEC/VALUE(7,3),RTABN(3,3),RTABR(3,3),ATABN(3),ATABR(3),
1SN,SS,WN,WS,RS,RW,NC,INRAND,VRANEF,CPN,CP,SP,SPAM,STRAT5,
2PNOLD,ISN,ISS,ECNXX,ECNXS,PN,PS,FACTXN,FACTXS

WRITE (6,1)
1 FORMAT (//,' S/R ECONOMY ERROR ************')
RETURN
END
SUBROUTINE FNAT(PTRUTH)

THE PURPOSE OF THIS SUBROUTINE IS TO DETERMINE PATTERNS IN THE
EN (NORTH'S ESTIMATE OF SOUTH'S RELIABILITY) ARRAY, IN ORDER TO
PREDICT THE TRUTHFULNESS OF SOUTH'S DECLARATION.

PTRUTH=0 ==> SOUTH'S DECLARATION IS PROBABLY NOT TRUE.
PTRUTH=1 ==> SOUTH'S DECLARATION IS PROBABLY TRUE.

IMPLICIT "INTEGER (A-W)
COMMEN/JEC/VALUE(7,3), RTABN(3,3), RTABS(3,3), ATABN(3), ATABS(3),
1SN, SS, WN, WS, RN, NC, INRAND, VRAND, EFLAG, CPN, CPS, STRATN, STRATS,
2PNOD, 1SN, ISS, ECNXN, ECNXS, PN, PS, FACTXN, FACTXS
COMMON/ENP/ENARRY(50), ESARRY(50)
COMMEN/EFF/EN,F, TPEN, TOPS
COMMEN/TA/AVAL4, AVAL3, AVAL2, NAME4(100), NAME3(100), NAME2(100)
COMMON/TA/TOP, ROTM, CA, CHAR, HIST, PLAYER, CATLAN, PATRNS, PATRNN
EQUIVALENCE (NAME4(4), RLINK4(3), LLINK4(2), DOWN4(1)), (NAME3(2), DOWN
12(1)), (NAME2(1), RLINK3(2), DOWN3(1))
COMMON/RLINK4(1000), LLINK4(1000), DOWN4(1000), RLINK2(1000), DOWN3
1(1000), DOWN2(1000)
C
L=1
IF (NC, GT, 4) GO TO 5
GO TO 1, 2, 3, 4, NC
1 CALL RAND
IF (VRAND, GT, 0.7) GO TO 101
C (ASSUME 70% CHANCE OF TELLING THE TRUTH FIRST TIME.)
100 PTRUTH=1
GO TO 99
101 PTRUTH=0
    GO TO 99
2 IF (NAME3(EN).EQ.50) GO TO 100
    PTRUTH=0
    GO TO 99
3 IF (NAME3(EN).EQ.50) GO TO 101
    IF (HE HAS TOLD TRUTH THE 1ST 2 TIMES, ASSUME HE WILL NOT 3RD.)
    PTRUTH=1
    GO TO 99
4 IF (NAME3(EN).EQ.50) GO TO 101
    IF (ENARY(2).EQ.50.AND.ENARY(3).EQ.50) GO TO 101
    ASSUME PATTERN OF 1-2-1 UNTRUTH/TRUTH/UNTRUTH.
    PTRUTH=1
    GO TO 99
C 5 IF (NAME3(EN).EQ.50) GO TO 100
    (HAS HE ALWAYS TOLD TRUTH?)
    N=NC-1
    IF (ENARY(N).EQ.501 GO TO 500
    (IF TOLD TRUTH LAST TIME, DETERMINE PATTERN.)
    DO 6 I=2,N
    IF (ENARY(NC-I).EQ.50) GO TO 75
    (IF HE HAS EVER TOLD TRUTH, DETERMINE UNTRUTH PATTERN.)
    CONTINUE
    PTRUTH=0
    GO TO 99
C 6 J=NC-1
    DO 76 I=J,N
    IF (ENARY(NC-I).LT.50) GO TO 77
    CONTINUE
    PTRUTH=1
    (HISTORY OF COMPLETE TRUTHFULNESS PRIOR LAST UNTRUTH, CAN NOT
    DETERMINE UNTRUTH PATTERN, ASSUME TRUTH.)
    GO TO 99
C 75 K=I-1
    DO 77 K=K,N
    IF (ENARY(NC-I).EQ.50) GO TO 79
    (K IS TRUTH PATTERN.)
    CONTINUE
    PTRUTH=0
2    (VERY POOR HONESTY DEMONSTRATED, ASSUME UNTRUTH.)
C 76 L=I-J+1
    L IS UNTRUTH PATTERN.
    (PATTERN IS APPROXIMATELY K/L/K TRUTH/UNTRUTH/TRUTH.)
    I=J-1
    IF (J.GE.1) GO TO 100
    GO TO 99
**SUBROUTINE TO BUILD THE LIBRARY OF PATTERNS SUPPORTS THE LEARNING CAPABILITIES OF THE PROGRAM.***

**THE TWENTY-FIVE SUBROUTINES WITHIN THIS SECTION.**

**IMPORTANT: DO NOT REMOVE OR ALTER THE ORDER OF THE STRINGS.**

**INPUT:** INTEGER (A,M)

**DESCRIPTION:**
- COMMON/UNIT, MAT, LEN, MAX
- COMMON/CHAR, NCH, PHYS, MACH, NAME

**SUBROUTINE CATLOG**

**PRTRUTH = 0**

**GO TO 55**

**IF** (FNC-11 - L.T. 50) **GO TO 52**

**IF** (FNC-11 = L.T. 50) **GO TO 53**

**IF** (O.FNC-11.LT. 50) **GO TO 54**

**CONTINUE**

**PRTRUTH = 0**

**GO TO 55**

**CONTINUE**

**RETURN**
DIMENSION RLINK4(1000),LLINK4(1000),DOWN4(1000),RLINK3(1000),DOWN3(1000),DOWN2(1000)
EQUIVALENCE (NAME4(4),RLINK4(3),LLINK4(2),DOWN4(1)),(NAME2(2),DOWN2(1)),
(NAME3(3),RLINK3(2),DOWN3(1))
IF(CTLGN.EQ.0) GOTO 101
CATLN=CATLN+1
CALL GET3(L)
C
TOP=L
DOWN3(TOP)=0
RTHM=TOP
CATLN=TOP
NAME3(TOP)=PLAYER
GOTO 200
C
101 CALL SEARCH(PLY)
C CHECK TO SEE IF LISTED. IF IT IS MOVE PLAYER TO TOP OF CATLN.
IF(PLY.EQ.1)GOTO 200
C CHECK FOR 10 PLAYERS IN THE CATALOGUE. IF THERE IS DELETE THE
C BOTTOM PLAYER.
IF(CTLGN.EQ.10)GOTO 300
C
CATLN=CATLN+1
C
GET ADDITIONAL PLAYER CELL
C
400 TOP=CATLN
CALL GET3(L)
DOWN3(L)=TOP
TOP=L
CATLN=TOP
NAME3(TOP)=PLAYER
C
BUILD ADDITIONAL CELLS FOR EXPERIENCE LIBRARY ON NEW PLAYER.
C
50 CALL GET4(J)
C
CELL FOR CHARACTERISTICS HEADER.
RLINK3(TOP)=J
CHAR=J
C
CELL FOR AGGRESSIVENESS HEADER
CALL GET2(K)
RLINK4(CHAR)=K
AGGRS=K
NAME2(AGGRS)=0
DOWN3(AGGRS)=0
C
CELL FOR PARAMETERS
CALL GET4(J)
CALL GET4(J)
CELL FOR HISTORY HEADER
DOWN4(CHAR)=J
HIST=J
NAME4(HIST)=0
RLINK4(HIST)=0
LLINK4(HIST)=0
DOWN4(PARAP)=0

CALL GET2(K)
NAME4(CHAR)=K
ENPTR=K
NAME2(ENPTR)=0
DOWN2(ENPTR)=0
RETURN

GET SUBROUTINE TO DELETE A PLAYER FROM THE BOTTOM OF CATALOGUE.
300 CALL DELETP
GOTO 400

GET SUBROUTINE TO MOVE EXISTING PLAYER TO THE TOP OF THE CATALOGUE
200 CALL MOVEPL
RETURN

END

SUBROUTINE DELETP
SUBROUTINE TO DELETE A PLAYER FROM THE BOTTOM OF THE CATALOGUE.

IMPLICIT INTEGER (A-W)
COMMON/T/AVAILABLE,AVAILABLE2,NAMEA(1000),NAME2(1000),NAME3(1000)
COMMON/TO,TOP,BOTM,CATLG,CHAR,HIST,PLAYER,CATLN,PATRN,PATRNN
COMMON/TOP3/TYP3(20),INIT(20)
COMMON/TOP4/SEQ(20),SCONTR
DIMENSION RLINK(1000), LLINK(1000), DOWN(1000), RLINK3(1000), DOWN3(1000), DOWN3(1000), DOWN3(1000)  
EQUIVALENCE (NAME4(4), RLINK4(3), LLINK4(2), DOWN4(1)), (NAME2(2), DOWN2(1))  
IF(RLINK3(ROTM).EQ.0) GOTO 10  
RETURN  
CELL TO AVAIL2 LIST FROM ENPTR  
CHAR=RLINK3(ROTM)  
K2=NAME4(CHAR)  
DOWN2(K2)=RLINK4(CHAR)  
DOWN2(RLINK4(CHAR))=AVAIL2  
AVAIL2=K2  
RETURNS CELLS TO AVAIL4 FROM CHAR AND HIST.  
HIST=DOWN4(CHAR)  
CALL DSEQLB  
PARAM=LLINK4(CHAR)  
DOWN4(HIST)=PARAM  
DOWN4(PARAM)=AVAIL4  
AVAIL4=CHAR  
10 TOP=CATLG  
RETURNS PLAYER CELL TO AVAIL3  
11 IF(DOWN3(TOP).EQ.BOTM) GOTO 12  
TOP=DOWN3(TOP)  
GO TO 11  
12 DOWN3(BOTM)=AVAIL3  
AVAIL3=BOTM  
BOTM=TOP  
DOWN3(BOTM)=0  
RETURN  
END  

SUBROUTINE INLIB(INCNT, MOVE, TOPI, INCNTR, MOV1)  

INITIAL STRATEGY LIBRARY. THIS SUBROUTINE SETS UP CELLS FOR THE INITIAL STRATEGY LIBRARY OF CELLS.  
IMPLICIT INTEGER (A-H)  
COMMON/JEC/VALUE(7,3), RTABN(3,3), RTABS(3,3), ATABN(3), ATABS(3),  
SNN, SS, WNN, MS, RS, RN, NC, INRN, YRN, EFLAG, CPN, CPS, STRAT1, STRAT2,  
2FNDLD, I2SN, ISS, ECNXN, ECNXS, PN, PS, FACTN, FACTXS  
COMMON/TOAVAIL, AVAIL3, AVAIL2, NAME4(1000), NAME2(1000), NAME3(1000)  
COMMON/TD/TOP, BOTM, CATLG, CHAR, HIST, PLAYER, CATLG5, PATRNS, PATRNN
COMMON/TOP3/TYPE(20),INIT(20)
DIMENSION RLINK4(1000),LLINK4(1000),DOWN4(1000),RLINK3(1000),DOWN3
1(1000),DOWN2(1000)
EQUIVALENCE (NAME4(4),RLINK4(3),LLINK4(2),DOWN4(1)),(NAME2(2),DOWN
12(11)),(NAME3(3),RLINK3(2),DOWN3(11))
DATA A1/'A1'/,A2'/A2'/

"INCNT" IS THE COUNT OF CELLS IN THE LIBRARY, "INCNTR" IS THE
ALLOWABLE NUMBER OF CELLS IN THE LIBRARY. (IE, INITIALLY 10 CELLS
WERE THE MAXIMUM ALLOWED). "MOV" INDICATES WHETHER THE INITIAL
MOVES ARE IN THE LIBRARY OR IF A NEW CELL IS REQUIRED.

IF(INCNT.EQ.INCNTR) GOTO 10
INCNT=INCNTR+1
CALL GET4(J)
INIT(INCNT)=J
TOPI=J
RETURN

10 CALL DINTLB(INCNT,MOVE,TOPI)
RETURN

SUBROUTINE DINTLB(INCNT,MOVE,TOPI)

SUBROUTINE TO DELETE OR COMBINE SIMILAR ELEMENTS IN
THE INITIAL MOVE LIBRARY.

IMPLICIT INTEGER (A-W)
COMMON/JEC/VALUE(7,3),RTARN(3,3),RTABS(3,3),ATARN(3),ATABS(3),
1N,NN,N,P,RN,NC,IPR,AND,FLAG,CPN,CPS,STRAN,STRATS,
?PNU1C,TN,TCN,FCNXS,FS,FACNX,FACTS
COMMON/TA/AVAIL4,AVAIL3,AVAIL2,NAME4(1000),NAME2(1000),NAME3(1000)
COMMON/TD/TOP,ROT,M,CHAR,HIST,PLAYER,CATLGN,PATAMS,PATPNN
COMMON/TOP/MOVES,MOVEN
COMMON/TOP3/TYPE(20),INIT(20)
DIMENSION RLINK4(1000),LLINK4(1000),DOWN4(1000),RLINK3(1000),DOWN3
1(1000),DOWN2(1000)
EQUIVALENCE (NAME4(4),RLINK4(3),LLINK4(2),DOWN4(1)),(NAME2(2),DOWN
12(11)),(NAME3(3),RLINK3(2),DOWN3(11))
DATA A1/'A1'/,A2'/A2'/

C SUBROUTINE TO DELETE OR COMBINE INITIAL MOVES LIBRARY.
IF(MOVE.EQ.MOVEN) GOTO 20
IF(NAME4(HIST).EQ.0) GOTO 11
TOPI=NAME4(HIST)

8 VAL20=TOPI
RUNTOP=TOP

15 IF(RUNTOP.EQ.BOTH) GOTO 13
RUNTOP=DOWN3(RUNTOP)
CHARR=LINK3(RUNTOP)
HISTR=DOWN4(CHARR)
IF(MOVE.EQ.MOVEN) GOTO 22
IF(NAME4(HISTR).EQ.VAL20) GOTO 14
GOTO 15
13 TOPMN=MOVE
NAME4(TOP1)=NAME2(TOPMN)
TOPMN=DOWN2(TOPMN)
RLINK4(TOP1)=NAME2(TOPMN)
TOPMN=DOWN2(TOPMN)
RLINK4(TOP1)=NAME2(TOPMN)
DOWN4(TOP1)=0
RETURN
14 NAME4(HISTR)=0
GOTO 15
11 MCNT=1
12 TOP1=INIT(MCNT)
IF(MCNT.EQ.1) GOTO 100
RUNTOP=TOP
VAL20=TOP
GOTO 15
20 IF(RLINK4(HIST).EQ.0) GOTO 21
TOPI=RLINK4(HIST)
GOTO P
21 MCNT=2
GOTO 12
22 IF(RLINK4(HIST).EQ.VAL20) GOTO 24
GOTO 15
24 RLINK4(HIST)=0
GOTO 15
100 RETURN

SUBROUTINE TYPELIB(INCNT, WON, INCNT)
SUBROUTINE TO BUILD NEW STRUCTURES IN THE TYPE LIBRARY AT THE END OF A GAME.

IMPLICIT INTEGER (A-W)
COMMON/TOP7/GROUPS, GROUPN, TMoves, TMovEN, MCNT, CONT
COMMON/T/AVAIL4, AVAIL3, AVAIL2, NAME4(1000), NAME2(1000), NAME3(1000)
COMMON/T/TP4/ANT1, CATLN, CHAR, HIST, PLAYER, CATLN, PATRNS, PATTNN
COMMON/T/TP4/SEQ(7), SCNTR
COMMON/T/TP4/TYPE 201, INIT201
COMMON/T/TP8/RLH, ST, NAMEST
COMMON/CLS/DEFS, BDEFS, OFFEN, BOFFEN, PMOVEN, PMOVES
COMMON/TPJEC/DONOR,DOSQU,CKCNT1,CKCNT2
COMMON/JEC/VALUE(7,3),RTABN(3,3),RTAB2(3,3),ATABN(3),ATARS(3),
ISN,SS,WN,WS,RS,RIV,NC,INRND,VRND,EFLAG,CPN,CPS,STRATN,STRATS,
2PNOD,ISN,INS,ECNXN,ECNXS,PN,PS,FACTXN,FACTXS,
DIMENSION RLXN(1000),LLXN(1000),DOWN4(1000),RLXN3(1000),DOWN3(1000)
EQUIVALENCE (NAME4(4),RLXN(4),LXN4(4),DOWN4(1)), (NAME2(2),DOWN4(1)),
DATA A1/'A1'/,A2/'A2'/
CKCNT1=0
CKCNT2=0
DONOR=0
NAME4(HIST)=NAHIST
RLXN4(HIST)=RLHIST
IF(NC,LT.10)GOTO 101
IF(CONT.EQ.INCTR) GOTO 100
3 STRTP=LLXN4(PATRNS)
8 TOPP=DOWN4 STRTP)
RTOP=TOPGP
CONT=1
0 IF(CONT,EQ.INCTR) GOTO 500
IF(DOWN2(RTOP),EQ.0)GOTO 500
RTOP=DOWN2(RTOP)
CONT=CONT+1
GOTO 9
500 CONT=CONT+1
CALL GET3(L)
TYPE(CONTR)=L
TOPSN=L
RTOP1=RTOP
6 IF(DOWN2(RTOP1),EQ.0) GOTO 7
RTOP1=DOWN2(RTOP1)
GOTO 6
7 IF(DOWN2(RTOP1),EQ.0)GOTO 4
DOWN2(RTOP)=AVAIL2
AVAIL2=DOWN2(RTOP)
4 DOWN2(RTOP)=0
DOWN3 TOPSN=TOPGP
DOWN4 STRTP=0
C CHECK TO SEE IF BOTH NORTH AND SOUTH TYPE LIBRARY MEMORY IS SETUP.
IF(STRT4,EQ.LLHAIN(4,PATRNS))GOTO 50
TOPS=L
IF(DONOR12,2,102
2 STRTP=LLXN4(PATRNS)
GOTO 8
50 TOPSN=L
51 IF(WON)11,12,13
SUBROUTINE DTYPBL

SUBROUTINE TO DELETE OR COMBINE STRUCTURES IN THE TYPE LIBRARY.

IMPLICIT INTEGER (A-W)
COMMON/TOP4/SEQ(20), SCNTR
COMMON/TOP3/TYPE(20), INIT(20)
COMMON/JEC/VKUE(3, 3), ATABN(3, 31), ATABS(3, 3), ATABN(3), ATABR(3),
ISN, SS, WS, RS, RW, NC, INRAN, FNAND, EFLAG, CPN, CPS, STRATN, STRATS,
2PNOLD, ISN, ISS, ECNXN, ECNXS, PN, PS, FACTXN, FACTXS
COMMON/CSP/DEPS, DEFS, OFFFN, RPOFN, MOVES, PMOVES
COMMON/TOP/GROUPS, GROUPN, TMOVES, TMOVEN, MCNT, CONTR
COMMON/T/AVAIL3, AVAIL3, AVAIL2, NAME2(1000), NAME1(1000), NAME3(1000)
COMMON/T/TP, RTN, CATLG, CHAR, HIST, PLAYR, CATLGN, PATRN, PATMA
COMMON/TOP/MOVE, MOVEN
COMMON/T/TPJEC/DO充CR, OSDO, CKC1, CKC2
DIMENSION RLINK(1000), LLINK(1000), DOWN4(1000), RLINK3(1000), DOWN3
1(1000), DOWN2(1000)
EQUIVALENCE (NAME4(4), RLINK4(3), LLINK4(2), DOWN4(1)), (NAME2(2), DOWN
12(1)), (NAME3(3), RLINK3(2), DOWN3(1))
DATA A'/A1, A2/A2, PATRN
1 PATRN=PATRN
TOPI=NAME4(HIST)
2 FLAG=1
CONT=1
CONT2=1
STRP=LLINK4(PATRN)
MOVE=DOWN4(STRP)
TOPM=DOWN3(TYPE(CONT))
101 IF(DOWN2(TOPM), EQ, 0) GOTO 10
IF(DOWN4(MOVE), EQ, 0) GOTO 10
CKGP1=NAME2(TOPM)
CKGP2=NAME2(MOVE)
CALL CKYP(CKGP1, CKGP2, NGO, FLAG)
IF(NGO, EQ, 0) GOTO 20
FLAG=CONT2+1
CONT2=CONT2+1
CONT=CONT+1
GOTO 10
20 IF(CONT2, EQ, 4) GOTO 30
TOPM=DOWN2(TOPM)
MOVE=DOWN2(MOVE)
FLAG=1
GOTO 1C1
10 DOWN4(TOPI)=TYPE(CONT)
IF(PATRN, EQ, PATMA) GOTO 880
PATRN= PATRN
TOPI=RLINK4(HIST)
GOTO 2
AVAIL3=TOP1
MCT1=CONT
MCT2=MCT1+1
IF(MCT1.EQ.CONTR) GOTO 57
620 TYPE(MCT1)=TYPE(MCT2)
IF(MCT2.EQ.CONTR) GOTO 57
MCT1=MCT1+1
MCT2=MCT2+1
GOTO 620
57 CONTR=CONTR-1
IF(PATRN.EQ.PATRN) GOTO 570
CKCNT1=1
GOTO 572
570 CKCNT2=1
572 SCNT=1
58 TOPS=EQ(SCNT)
DTOPS=TOPS
DTOP1=DTOPS
59 DTOP2=DOWN3(DTOPS)
TOPT2=NAME3(DTOPS)
IF(TOPT,EQ.TOPT2) GOTO 64
DTOP1=DTOPS
IF(DOWN3(DTOPS).EQ.0) GOTO 65
DTOPS=DOWN3(DTOPS)
GOTO 56
65 IF(SCNT.EQ.SCNT) GOTO 66
SCNT=SCNT+1
GOTO 68
64 IF(DOWN3(DTOPS).EQ.0) GOTO 67
IF(DTOPS.EQ.TOPS) GOTO 68
DOWN3(DTOP1)=DTOP2
DOWN3(DTOPS)=RLINK3(DTOPS)
DOWN3(RLINK3(DTOPS))=AVAIL3
AVAIL3=DTOPS
DTOPS=DTOP2
GOTO 59
68 RTP=DTOP2
68 IF(DOWN3(DTOP2).EQ.0) GOTO 70
RTP=DTOP2
DTOP2=DOWN3(DTOP2)
GOTO 65
70 NAME3(DTOPS)=NAME3(DTOP2)
NAME3(RLINK3(DTOPS))=NAME3(RLINK3(DTOP2))
RLINK3(RLINK3(DTOPS))=RLINK3(RLINK3(DTOP2))
DOWN3(RLINK3(DTOPS))=DOWN3(RLINK3(DTOP2))
DOWN3(DTOP2)=DOWN3(DTOP2)
AVAIL3=DOWN3(RTP)
DOWN3(RTP)=0
GOTO 58
67 IF(DTOPS.EQ.TOPS) GOTO 71
DOWN3(DTOPS)=RLINK3(DTOPS)
DOWN3(RLINK3(DTOPS))=AVAIL3
AVAIL3=DOWN3(DTOP1)
DOWN3(DTOP1)=0
GOTO 65
71 RUNT=CATLG
72 CHARR=RLINK3(RUNT)
HISTR=DOWN4(CHARR)
720 IF(TOPS.EQ.LLINK4(HISTR)) GOTO 710
71 IF(RUNT.EQ.ROTM) GOTO 730
RUNT=DOWN3(RUNT)
GOTO 72
710 LLINK4(HISTR)=0
GOTO 720C
72C DOWN4(HISTR)=0
GOTO 73
730 DOWN3(DTOPS)=RLINK3(DTOPS)
DOWN3(RLINK3(DTOPS))=AVAIL3
AVAIL3=DOWN3(DTOPS)
CT1=SCNT
C1:=SCNT+1
IF(CT1.EQ.SCONTR) GOTO 80
731 SEOH(CT1)=SEOH(CT2)
IF(CT2.EQ.SCONTR) GOTO 80
CT1=CT1+1
CT2=CT2+1
GOTO 731
80 SCONTR=SCONTR-1
GOTO 65
890 CT=CT+1
RETURN
66 IF(PATRN.EQ.PATRN) GOTO 660
PATRN=PATRN
GOTO 2
66C RETURN
END

SUBROUTINE SEQLB(INCNR,WON)
SUBROUTINE TO BUILD THE SEQUENCE LIBRARY.
IMPLICIT INTEGER (A-W)
COMMON/T/AVAIL4,AVAIL3,AVAIL2,NAM4(1000),NAM2(1000),NAM3(1000)
COMMON TD/TOP, BOTM, CATLC, CHAR, HIST, PLAYER, CATLGN, PATRNS, PATRNN
COMMON TOP/STK(10), STRM(10), STKN(10), STKNM(10)
COMMON TOP3/TYP(20), INITI(20)
COMMON TP4/SEQ(20), SCONTR
COMMON TPD7/GROUPN, TMOVES, TMOVEN, MCNT, CCNTR
COMMON JEC/VALUE(7, 3), PTABN(3, 3), PTABS(3, 3), ATABN(3), ATABS(3),
1 SN, SS, WN, WS, RS, RW, NC, INRAND, YRAND, EELAC, CPN, CPS, STRATN, STRATS,
2PHOLD, 1SN, ISS, ECNXX, ECNX, PN, PS, FACTXN, FACTXS
DIMENSION RLINK4(1000), LLINK4(1000), DWN4(1000), RLINK4(1000), DWN3(1000)
11 = (NAME3(4), RLINK4(3), LLLNK4(2), DWN4(1)), (NAME3(2), DWN4(1))
DATA A1'A1',A2'A2,'
PSN-1
IF(NC.LT.10)GOTO 401
CNT=1
PCN=0
SCNX=INCNTR+10
14 IF(SCNX.GT.SCNX)GOTO 100
15 IF(LINK4(HIST).EQ.0)GOTO 399
TPSO=LLNK4(HIST)
12 IF(NAME3(HIST).EQ.0)GOTO 100
13 IF(NAME3(RTOP).EQ.TYPENO)GOTO 400
16 IF(DWN3(RTOP).EQ.0)GOTO 100
17 IF(NAME3(RTOP).EQ.0)GOTO 100
18 IF(NAME3(RTOP).EQ.0)GOTO 100
19 IF(CNT.LE.INCNTR)GOTO 100
CALL GET3(L)
C
10 DWN3(RTOP)=L
11 RTOP=L
12 DWN3(RTOP)=0
13 NAME3(RTOP)=TYPENO
14 CALL GET3(L)
15 LINK3(RTOP)=L
16 MOVES=L
17 NAME3(IMOVES)=NAME4(INITS)
18 LINK3(IMOVES)=LINK4(INITS)
19 DWN3(IMOVES)=L
20 GOTO 100
395 INITS=NAME4(HIST)
400 TYPENO=DOWN4(INITS)
GOTO 104
**SUBROUTINE DSEOLB**

SUBROUTINE TO DELETE THE SEQUENCE STRUCTURES OF A PLAYER AND THE SEQUENCE STRUCTURES FOR THE ASSOCIATED COMPUTER PATTERNS OF PLAY.

**IMPLICIT INTEGER (A-H)**

**COMMON/IT/AVAIL, AVAIL3, AVAIL2, NAME4(1000), NAME2(1000), NAME3(1000)**

**COMMON/TOP, TOPM, CATLG, CHAR, HIST, PLAYER, CATLGN, PATRNS, PATRNM**

**COMMON/TOP2, STKM(10), STK(10), STKM(10), STK(10)**

**COMMON/TOP4/SEQ(20), SCONTR**
COMMON/TDP3/TYPE(20), INIT(20)
COMMON/JEC/VALUE(7,3), RTABN(3,3), ATARN(3), ATARS(3),
1SN, SS, WN, WS, RS, RW, NC, INRAND, YR AND, EFLAG, CP, CPS, STRATN, STRATS,
2PNOLD, ISN, ISSN, ECNXN, ECNXS, PN, PS, FEXTN, FACTXS
COMMON/CLS/DEFES, OFFEN, ROFFEN, PMOVEN, PMOVES
COMMON/TOP7/GROUPS, GROUPN, TMOVES, TMOVEN, MCNT, CONTR
DIMENSION RLINK4(1000), LLINK4(1000), DOWN4(1000), RLZ4N(1000), DOWUS
1(1000), DOWUS1(1000)
EQUIVALENCE (NAME4(4), RLINK4(3), LLINK4(2), DOWN4(1)), (NAME2(2), DOWN
12(1)), (NAME3(3), RLINK3(3), LLINK3(2), DOWN3(1))
DATA A1/'A11', A2/'A21'/
IF(LLINK4(HIST)) 7, 7, 8
TOPS=LLINK4(HIST)
8 TOPS=TOPS
10 DTP1=TOPS
IF(DOWN3(DTP1).EQ.0) GOTO 11
IF(DTP1.EQ.TOPS) GOTO 10
DTP1=DOWN3(DTP1)
DTP1=DOWN3(DTP1)
DTP1=DOWN3(DTP1)
DTP1=DOWN3(DTP1)
GOTO 2
11 DTP1=LLINK3(DTP1)
DTP1=DOWN3(DTP1)
DTP1=AVAIL3
AVAIL3=TOPS
IF(TOP5.EQ.DOWN4(HIST)) GOTO 20
IF(DOWN4(HIST)) 20, 20, 9
TOPS=DOWN4(HIST)
GOTO 10
20 RETURN
END

SUBROUTINE PATRN
SUBROUTINE TO BUILD THE TEMPORARY MEMORY HEADER NODES FOR
 BOTH THE PLAYER (PATRNS) AND THE COMPUTER (PATRNN).

IMPLICIT INTEGER (A-N)
COMMON/T/AVAIL3, AVAIL4, NAME4(1000), NAME3(1000)
COMMON/CLS/DEFES, OFFEN, ROFFEN, PMOVEN, PMOVES
COMMON/JEC/VALUE(7,3), RTABN3(3,3), ATARN(3), ATARS(3),
1SN, SS, WN, WS, RS, RW, NC, INRAND, YR AND, EFLAG, CP, CPS, STRATN, STRATS,
2PNOLD, ISN, ISSN, ECNXN, ECNXS, PN, PS, FEXTN, FACTXS
COMMON/TOP7/GROUPS, GROUPN, TMOVES, TMOVEN, MCNT, CONTR
DIMENSION RLINK4(1000),LLINK4(1000),DOWN4(1000),RLINK3(1000),DOWN3
1(1000),DOWN2(1000)
EQUIVALENCE (NAME4(4),RLINK4(3),LLINK4(2),DOWN4(1)),(NAME2(2),DOWN
12(11)),(NAME3(3),RLINK3(2),DOWN3(1))
DATA A1/'A1'/,A2/'A2'/
PATRNS=1
PATTERN=1
CALL GET4(J)

GET HEADER CELL FOR PATTERN MEMORY OF SOUTH.

PATRNS=J
TOPS=J
NAME4(TOPS)=0
PLINK4(TOPS)=0

GET HEADER CELL FOR GROUP MOVES THIS GAME FOR SOUTH

CALL GET4(J)
LLINK4(TOPS)=J
STRTP=J
NAME4(STRTP)=0
PLINK4(STRTP)=0
DOWN4(STRTP)=0
DOWN4(TOPS)=0
CALL GET4(J)

GET HEADER CELL FOR PATTERN MEMORY OF NORTH.

PATTERN=J
TOPN=J
NAME4(TOPN)=0
RLINK4(TOPN)=0

GET HEADER CELL FOR GROUP MOVES THIS GAME FOR NORTH

CALL GET4(J)
LLINK4(TOPN)=J
STRTPN=J
NAME4(STRTPN)=0
RLINK4(STRTPN)=0
LLINK4(STRTPN)=0
DOWN4(STRTPN)=0
DOWN4(TOPN)=0
RETURN
END
SUBROUTINE DPATHS

SUBROUTINE TO DELETE THE REMAINING NODES IN TEMPORARY MEMORY AT THE END OF THE GAME.

IMPLICIT INTEGER (A-W)
COMMON/T/D/AVAIL4,AVAIL3,AVAIL2,NAME4(1000),NAME2(1000),NAME3(1000)
COMMON/T/D/TOP,CTMG,CHAR,HIST,PLAYER,CTLMN,PTMPNS,PTMPN
COMMON/TOP2/STR(10),STKM(10),STKN(10),STKNM(10)
DIMENSION RLINK4(1000),LLINK4(1000),DOWN4(1000),RLINK3(1000),DOWN3
1(1000),DOWN2(1000)
EQUIVALENCE (NAME4(1),RLINK4(3),LLINK4(2),DOWN4(11),(NAME2(21),DOWN
12(1)),(NAME3(3),RLINK3(32),DOWN3(11))
WRITE(6,5)
5 FORMAT('D PATNS *)
PATRN=PTMPNS
20 STRP=LLINK4(PATRN)
IF(DOWN4(STRTP),EQ.0)GOTO 10
TOPGP=DOWN4(STRTP)
12 IF(DOWN2(TOPGP),EQ.0)GOTO 11
TOPGP=DOWN2(TOPGP)
GOTO 12
11 DOWN2(TOPGP)=AVAIL2
AVAIL2=DOWN4(STRTP)
DOWN4(STRTP)=0
10 IF(DOWN4(PATRN),EQ.0)GOTO 13
MOVE=DOWN4(PATRN)
15 IF(DOWN2(MOVE),EQ.0)GOTO 14
MOVE=DOWN2(MOVE)
GOTO 15
14 DOWN2(MOVE)=AVAIL2
AVAIL2=DOWN4(PATRN)
DOWN4(PATRN)=STRTP
DOWN4(STRTP)=AVAIL4
AVAIL4=PTMPN
13 IF(PATRN,EQ.PTMPN)GOTO 55
PTMPN=PTMPN
GOTO 20
55 RETURN
END

SUBROUTINE PATUPD

SUBROUTINE TO UPDATE TEMPORARY MEMORY OF BOTH NORTH (PATRN)
AND SOUTH (PTMPN)
SUBROUTINE DPATNS
SUBROUTINE TO DELETE THE REMAINING NODES IN TEMPORARY
MEMORY AT THE END OF THE GAME.

IMPLICIT INTEGER (A-H)
COMMON/T,AVAIL4,AVAIL3,AVAIL2,NAM4(1000),NAME2(1000),NAME3(1000)
COMMON/TOP,TOP2,BOTH,CATLG,CHR,HIST,PLAYER,CATLN,PATRNS,PATRN
COMMON/TOP2/STRM(10),STRM1(10),STKN1(10),STKNM(10)
DIMENSION RLINK4(1000),RLINK3(1000),RLINK2(1000),DOWN3
1(1000),DOWN2(1000)
EQUIVALENCE (NAME4(4),PLINK4(3),LLINK4(2),DOWN4(11),(NAME2(2),DOWN
12(1)),(NAME3(3),RLLINK3(2),DOWN3(1))
WRITE(4,5)
5 FORMAT(1'DPATNS ',/)
   PATRN=PATRNS
20 STRTP=LLINK4(PATRN)
   IF(DOWN4(STRTP).EQ.0)GOTO 10
   TOPCP=DOWN4(STRTP)
10   DOWN2(TOPCP)=AVAIL2
   AVAIL2=DOWN4(STRTP)
   DOWN4(STRTP)=0
   IF(DOWN4(PATRN).EQ.0)GOTO 13
   MOVE=DOWN4(PATRN)
12   IF(DOWN2(MOVE).EQ.0)GOTO 14
   MOVE=DOWN2(MOVE)
13   DOWN2(MOVE)=AVAIL2
   AVAIL2=DOWN4(PATRN)
   DOWN4(PATRN)=STRTP
   DOWN4(STRTP)=AVAIL4
   AVAIL4=PATRN
14   IF(PATRN.EQ.PATRN)GOTO 55
   PATRN=PATRNS
GOTO 20
55 RETURN
END
SUBROUTINE PATUPD
SUBROUTINE TO UPDATE TEMPORARY MEMORY OF BOTH NORTH (PATRN)
AND SOUTH (PATRNS)
IMPLICIT INTEGER (A----W)
COMMON/T(AVAIL4,AVAIL3,AVAIL2,NAME4(1000),NAME2(1000),NAME3(1000)
COMMON/TOP,ROTMC,CHAR,MCNT,PLAYER,CATLN,PLAYER,CATLN,PLAYER,CATLN
COMMONE/TYPE(20),INIT(20)
COMMON/MOVES,MOTION
COMMON/GROUPS,GROUPN,TM0VES,TM0VEN,MCNT,CONT0P
COMMON/DEFES,DEFF,OPEN,OPEN,PM0VEN,PM0VES
COMMON/JEC/VALUE(7,3),RTABN(3,3),RTABLE(3,3),ATABN(3,3),ATABLE(3,3),
ISN,SS,WS,RS,RM,NC,INRAND,YRAND,FLAG,CPS,STRATN,STRATS,
2NOLD,ISN,ISS,ECXN,ECXN,PS,FCTXN,FACTXS
D 0 MENS I0N RLINKA(1000),LLINKA(1000),DOWNA(1000),RLINK3(1000),DOWN3
11000),DOWN4(1000)
EQUIVALENC E (NAME4(4),RLINK4(3),LLINK4(2),DONW4(1)),(NAME2(2),DOWN
12(1)),(NAME3(3),RLINK3(2),DONW3(1))
TOPN=PLAYER
TOPS=PLAYER
IF(NC.GT.3) GOTO 1
IF(DOWN4(TOP$),EQ.0) GOTO 10
TOPS=DOWN4(TOP$)
1
IF(DOWN2(TOP$),EQ.0) GOTO 5
TOPS=DOWN2(TOP$)
10
CALL GET2(K)
DOWN2(TOP$)=K
TOPS=DOWN2(TOP$)
NAME2(TOP$)=STRATS
DOWN2(TOP$)=0
STROAT=1
CALL STRATOT(STROAT)
IF(NE.GT.3) GOTO 4
GOTO 20
10
CALL GET2(K)
DOWN4(TOP$)=K
MOVES=K
TOPS=MOVES
DOWN2(TOP$)=0
NAME2(TOP$)=STRATS
STROAT=1
CALL STRATOT(STROAT)
GOTO 20
4
TOPS=MOVES
MOVES=MOVES
BYPASS=0
C
TOTAL VALUE OF THREE MOVES TO DETERMINE THE GROUP THEY BELONG TO.
CALL TOTAL(MOVE,TOTRE,TRUE,BYPASS,INCNT,MCNT,TOTAL,TOTAL2)
TM0VES=TOTAL2
CALL TOTMV(TOTAL, GROUP)
STRTP=LLINK4(TOPS)
TOPGPS=STRTP
IF(DOWN4(TOPGPS).EQ.0)GOTO 25
TOPGPS=DOWN4(TOPGPS)
27 IF(DOWN2(TOPGPS).EQ.0)GOTO 26
TOPGPS=DOWN2(TOPGPS)
GOTO 27
25 CALL GET2(K)
DOWN4(TOPGPS)=K
TOPGPS=K
DOWN2(TOPGPS)=0
NAME2(TOPGPS)=GROUP
GOTO 26
26 CALL GET2(K)
DOWN2(TOPGPS)=K
TOPGPS=K
NAME2(TOPGPS)=GROUP
DOWN2(TOPGPS)=0
GOTO 20
DELETE OLDEST OF THREE MOVES ADD LAST MOVE TO BOTH NORTH AND SOUTH
1 TOPMS=MOVES
TOPMN=MOVES
31 PT=DOWN2(TOPMS)
NAME2(TOPMS)=NAME2(PT)
IF(DOWN2(PT).EQ.0) GOTO 30
TOPMS=DOWN2(TOPMS)
GOTO 31
30 NAME2(PT)=STRATS
STRAT=1
CALL STRTOT(STRAT)
SHIFT NEW MOVE INTO BOTTOM OF NORTH'S MOVE PATTERN AND UPDATE TOTAL
HEADER.
32 PT=DOWN2(TOPMN)
NAME2(TOPMN)=NAME2(PT)
IF(DOWN2(PT).EQ.0) GOTO 35
TOPMN=DOWN2(TOPMN)
GOTO 32
35 NAME2(PT)=STRATN
STRAT=0
CALL STRTOT(STRAT)
GOTO 4
20 IF(NC.GT.3) GOTO 104
IF(DOWN4(TOPMN).EQ.0) GOTO 110
TOPMN=DOWN4(TOPMN)
108 IF(DOWN2(TOPMN).EQ.0) GOTO 105
TOPMN=DOWN2(TOPMN)
GOTO 108
105 CALL GET2(K)
DOWN2(TOPMN)=K
TOPMN=DOWN2(TOPMN)
DOWN2(TOPMN)=0
NAME2(TOPMN)=STRATN
STRAT=C
CALL STRTOT(STRAT)
IF(NC.EQ.3) GOTO 104
RETURN
110 CALL GET2(K)
DOWN4(TOPN)=K
MOVEN=MOVEN
DOWN2(TOPMN)=0
NAME2(TOPMN)=STRATN
STRAT=0
CALL STRTOT(STRAT)
RETURN
104 MOVEN=DOWN4(PATRNN)
MOVE=MOVEN
BYPASS=0
CALL TOTALM(MOVE,TOTTRUE,BYPASS,TMCNT,MCNT,TOTAL,TOTAL2)
MOVEN=TOTAL2
CALL TOTMV(TOTAL,GROUP)
STRPN=LINK4(TOPN)
TOPGPN=STRPN
IF(DOWN4(TOPGPN).EQ.0) GOTO 125
TOPGPN=DOWN4(TOPGPN)
127 IF(DOWN2(TOPGPN).EQ.0) GOTO 126
TOPGPN=DOWN2(TOPGPN)
GOTO 127
125 CALL GET2(K)
DOWN4(TOPGPN)=K
TOPGPN=K
DOWN2(TOPGPN)=0
NAME2(TOPGPN)=GROUP
RETURN
126 CALL GET2(K)
DOWN2(TOPGPN)=K
TOPGPN=K
NAME2(TOPGPN)=GROUP
DOWN2(TOPGPN)=0
RETURN
END
SUBROUTINE REMEMR(INCNT, INCNT)

SUBROUTINE TO RECONSTRUCT PERMANENT MEMORY PRIOR TO
A NEW GAME OR SERIES OF GAMES.

IMPLICIT INTEGER (A-W)
COMMON/TOPE/APN,APS,STCK(40)
COMMON/PARP(4),S(4),DBLS(6)
COMMON/FEEN,ES,TOPN,TOPE
COMMON/ENP/ENARRY(50),ENARRY(50)
COMMON/CLEN/DEPES,BOPE,OFFN,POFFN,PMOVEN,PMOVES
COMMON/JEC/VALUE(7,3),RTABN(3,3),RTABN(3,3),ATABN(3,3),ATABN(3,3),
ISN,SS,WN,WS,RS,RN,NC,INRAND,VRAND,EFALL,CPN,PS,STRATN,STRATS,
2PNLD,IGN,TS,FCNKN,FCNKS,PN,PS,FACTXY,FACTXS
COMMON/ST/STRN,LASTG,FLAG,TURN, ANAL, RECON
COMMON/T/AVAIL1,AVAIL2,AVAIL2,NAME2(1000),NAME2(1000),NAME2(1000)
COMMON/T/TOP,TOTM,CAUL,CHAR,HIST,PLAYER,CAULN,PATRNS,PATRNN
COMMON/T/TOP2/STK(I0),STKN(I0),STKNM(I0)
COMMON/T/TOP3/TYPE(20),INIT(20)
COMMON/T/TOP4/SEQ(20),SCNTR
COMMON/T/TOP7/GROUPS, GROUPN, TMOVES, TMOVEN, MCNT, CONTR
DIMENSION COMPN(3,3)
DIMENSION RLINK4(1000), LLINK4(1000), DLINK4(1000), RLINK3(1000), DLINK3(1000),
DLINK2(1000), DLINK2(1000), DLINK2(1000), (NAME2(12), DLINK2(12), DLINK2(12), DLINK2(12),
DATA HEADS/TAIL, TAIL/TAIL, YES/YES, NO/NO/
DATA A1/'A1/A2/A2/A2'
WRITE(6,137)
137 FORMAT (1 DO YOU WISH TO READ MEMORY AND REBUILD THE HISTORY OF
PLAYERS, YES OR NO "A3" FORMAT RIGHT JUSTIFIED)
READ(5,655)DU
695 FORMAT (A3)
IF (DU.EQ.NO) GOTO 27
READ(2,459)TSTK,(STACK(I),I=1,TSTK)
455 FORMAT(9I3)
TSTK(1),TSTK(2),TSTK(3),TSTK(4),TSTK(5),TSTK(6),TSTK(7),TSTK(8),TSTK(9)
READ(2,101)RBLD, INCNT, INCNTR, CONTR, SCONTR, APN, APS
101 FORMAT((7I3)
CALL REBUILD(INCNT)
CATLGN=0
10 IF (CATLGN.EQ.NO) GOTO 22
READ(2,102)END, (PLAYER, NEM, DEM, NAG, DAG, NAP, RPA, LPA, DPA, NHIS,
PHIS, LIS, HISS)
102 FORMAT(44I12I3)
CALL CATLOG
ENPTR=NAME4(CHAR)
AGGRS=RLINK4(CHAR)
PARAM=LLINK4(CHAR)
NAME2(ENPTR)=NEM
DOWN2(ENPTR)=DEM
NAME2(AGRSS)=NAG
DOWN2(AGRSS)=DAG
NAME4(PARAM)=NAP
RLINK4(PARAM)=RPA
LLINK4(PARAM)=LPA
DOWN4(PARAM)=DPA
NAME4(HIST)=NHS
RLINK4(HIST)=HST
LLINK4(HIST)=LHS
DOWN4(HIST)=DHS
GOTO 10

50  CONTINUE
21  MCNT=0
22  IF(MCNT.EQ.INCNT)GOTO 23
23  TCONT=0
24  IF(TCONT.EQ.NEXPR)GOTO 230
25  TCONT=0
26  IF(TCONT.EQ.NEXPR)GOTO 230
104  FORMAT(414)
105  MCNT=MCNT+1
106  TOPI=INIT(MCNT)
107  NAME4(TOPI)=MOVE1
108  RLINK4(TOPI)=MOVE2
109  LLINK4(TOPI)=MOVE3
110  DOWN4(TOPI)=TYPE
111  GOTO 21
20  MCT=0
21  MCNT=0
22  IF(MCNT.EQ.INCNT)GOTO 23
23  TCONT=0
24  IF(TCONT.EQ.NEXPR)GOTO 230
25  TCONT=0
26  IF(TCONT.EQ.NEXPR)GOTO 230
104  FORMAT(414)
105  MCNT=MCNT+1
106  TOPT=TYPE(TCONT)
107  NAME2(TOPT)=OFF
108  RLINK2(TOPT)=OFF
109  CALL GET2(J)
210  DOWN3(TOPT)=J
211  TTOP=J
22  READ(2,100)NAME2,DOWNR
23  FORMAT(214)
24  IF(NAME2.EQ.0.15.OR.NAME2.EQ.0.25)GOTO 580
25  NAME2(TOPG)=NAME2
250  IF(DOWN2.EQ.0)GOTO 25
26  CALL GET2(J)
27  DOWN2(TOPG)=J
28  TTOP=DOWN2(TOPG)
29  GOTO 24
30  CALL GET2(J)
31  NAME2(TOPG)=A1
32  NAME2(TOPG)=A2
33  NAME2(TOPG)=A3
34  NAME2(TOPG)=A4
35  NAME2(TOPG)=A5
36  NAME2(TOPG)=A6
37  NAME2(TOPG)=A7
38  NAME2(TOPG)=A8
39  NAME2(TOPG)=A9
40  NAME2(TOPG)=A10
SUBLROUTINE REBUILD(INCNT)

SUBLROUTINE TO REBUILD THE HEADER CELLS FOR THE INITIAL, TYPE, AND SEQUENCE LIBRARIES.

IMPLICIT INTEGER (A-W)
COMMON/TOJ/E/PN,APS,STACK(40)
COMMON/PAR/P(4),S(4),PRLS(5)
COMMON/ELEN,E,TMP,E,NAP
COMMON/ENP,ENPARY(50),ESARRY(50)
COMMON/CLS/DEFS,BDEFS,DEFEN,REDEN,PMOVEN,PMOVES
COMMON/JC/VALUE(7,3),RTABN(3,3),RTARS(3,3),ATABN(3),ATARS(3)
COMMON/ST/STRE,LASTG0,NFLAG,TURN,ANAL,RECN
COMMON/TH/AVAILABLE,AVAILABLE,AVAILABLE,NAME2(1000),NAME2(1000),NAME3(1000)
COMMON/TO/TOP,STRN,CHAR,STRN,PLAYER,CHAR,PLAYER,CHAR,PLAYER
COMMON/TP/TOP/TOP/TYPEP,TYPEN,TYPEN,TYPEN,TYPEN,TYPEN
COMMON/T finances, finance, finances,
COMMON/TDP3/TYPE(20),INIT(20)
COMMON/TDP4/SEQ(20),SCONTR
COMMON/TDP7/GROUPS,GRUPF,TMOVES,TMOVEN,MCNT,CONTR
DIMENSION COMPN(3,3)
DIMENSION RLINK4(1000),L2LINK4(1000),DOWN4(1000),RLINK3(1000),DOWN3
1(1000),DOWN2(1000)
EQUIVALENCE (NAME4(3),RLINK4(3),L2LINK4(2),DOWN4(11),NAME2(2),DOWN
12(11)),(NAME3(3),RLINK3(2),DOWN3(1))
DATA HEADS/HEAD/,TAILS/'TAIL'/,YES/'YES'/,NO/'NO'/
DATA A1/A1'/A2'/A2'/'
RSTK=0
SSTK=0
TSTK=0
11 IF(TSTK.EQ.INCNT) GOTO 40
TSTK=TSTK+1
41 TJ=AVAIL4
JJ=TJ
42 CALL GETE(J)
IF(STACK(TSTK).EQ.J) GOTO 15
PJ=JJ
DOWN4(JJ)=AVAIL4
JJ=DOWN4(JJ)
GOTO 42
15 INIT(TSTK)=J
DOWN4(RJ)=AVAIL4
AVAIL4=TJ
DOWN4(J)=0
GOTO 11
40 IF(RSTK.EQ.CONTR) GOTO 400
RSTK=RSTK+1
TSTK=TSTK+1
TL=AVAIL3
LL=TL
410 CALL GETE(L)
IF(STACK(TSTK).EQ.L) GOTO 150
RL=LL
DOWN3(LL)=AVAIL3
LL=DOWN3(LL)
GOTO 410
150 TYPE(RSTK)=L
DOWN3(RL)=AVAIL3
AVAIL3=TL
GOTO 40
450 IF(SSTK.EQ.SCONTR) GOTO 450
SSTK=SSTK+1
TSTK=TSTK+1
TL=AVAIL3
LL=TL
STACK(TSTK)=SEO(SSTK)
GOTO 500

TSTK=TSTK+1
STACK(TSTK)=0
WRITE(3,601)TSTK,(STACK(I),I=1,TSTK)
WRITE(3,101)CATLGN,INCNT,INCNTR,CONTR,SCONTR,APN,APS

FORMAT(13,2013)
TOP=ROTM
B0T=R0TM
DO 20 I=1,CATLGN
CHT=RLINK3(TOP)
ENPTR=NAME4(CHT)
AGGRS=RLINK4(CHT)
PAPAR=LLINK4(CHT)
WRITE(3,102)NAME3(TOP),NAME2(ENPTR),DOWN2(ENPTR),
1NAME2(AGGRS),DOWN2(AGGRS),NAME4(PAPAR),RLINK4(PAPAR),
2LLINK4(PAPAR),DOWN4(PAPAR),NAME4(HIST),RLINK4(HIST),
3LLINK4(HIST),DOWN4(HIST)

FORMAT(A4,1213)
IF(DU.EQ.NO)GOTO 55
WRITE(6,103)NAME3(TOP),NAME2(ENPTR),DOWN2(ENPTR),
1NAME2(AGGRS),DOWN2(AGGRS),NAME4(HIST),RLINK4(HIST),LLINK4(HIST),
2DOWN4(HIST),NAME4(PAPAR),RLINK4(PAPAR),LLINK4(PAPAR),DOWN4(PAPAR)
55 IF(TOP.EQ.CATLG)GOTO 99
TOP=CATLG
551 IF(DOWN3(TOP),EQ.R0TM)GOTO 550
TOP=DOWN3(TOP)
GOTO 551
550 R0TM=TOP
20 CONTINUE
96 DO 22 I=1,INCNT
TOP1=INIT(I)
MOVE1=NAME4(TOP1)
MOVE2=RLINK4(TOP1)
MOVE3=LLINK4(TOP1)
TYPE=DOWN4(TOP1)
WRITE(3,104)MOVE1,MOVE2,MOVE3,TYPER

FORMAT(414)
IF(DU.EQ.NO)GOTO 56
WRITE(6,105)INIT(I),MOVE1,MOVE2,MOVE3,TYPER
56 RUN=1
22 CONTINUE
DO 23 I=1,CONTR
TOPT=TYPE(I)
DEF=NAME3(TOPT)
G=0
OFF=RLINK2(TOPT)
106 WRITE(3,106)DEF,OFF
107 FORMAT(214)
108 IF(DU.EQ.NO)GOTO 59
109 WRITE(6,112)1,TOP,T,DEF,OFF
110 FORMAT(5X,'TYPE','I2','I3','/2X','DEF=','I3','/2X','OFF=','I3')
111 TOP=DOWN3(TOP)
55 IF(NAME2(RTOPT),EQ,11)OR,NAME2(RTOPT),EQ,2)GOTO 60
112 WRITE(3,17)NAME2(RTOPT),DOWN2(RTOPT)
113 FORMAT(214)
114 IF(DU.EQ.NO)GOTO 58
115 WRITE(6,111)G,NAME2(RTOPT)
116 FORMAT(5X,'GROUP','I2','I3','I4')
117 IF(DOWN2(RTOPT),EQ,0)GOTO 23
118 RTOPT=DOWN2(RTOPT)
119 GOTO 57
60 IF(NAME2(RTOPT),EQ,11)GOTO 633
61 IF(DU.EQ.NO)GOTO 580
62 WRITE(6,110)G,NAME2(RTOPT)
63 FORMAT(214)
64 IF(DU.EQ.NO)GOTO 590
65 WRITE(6,110)G,NAME2(RTOPT)
66 FORMAT(5X,'GROUP','I2','I3','I4')
67 CONTINUE
68 DO 24 T=1,SCONT
69 FLAG=0
70 TOP=SEQ(1)
71 RTOPS=TOPS
72 TYPES=NAME3(RTOPS)
73 MOVI=RLINK3(RTOPS)
74 MOVII=NAME3(MOVI)
75 MOVI2=RLINK3(MOVI)
76 MOVII=DOWN3(MOVI)
77 IF(FLAG.GT.0)GOTO 125
78 DOWN=DOWN3(RTOPS)
79 WRITE(3,121)TYPES,MOVII,MOVI2,MOVI3,DOWN
80 IF(DU.EQ.NO)GOTO 24
81 IF(FLAG.GT.0)GOTO 24
82 WRITE(6,123)TOPS
FLAGR=1
GOTO 117
125 WRITE(6,124) TYPES, MOVIT, MOV12, MOV3
GOTO 116
24 CONTINUE
103 FORMAT(5X,'PLAYER ',A4,2X,'ENAVG',I4,2X,'ESAVG',I4,2X,'AGG',I4,2X,'SEQ',I4,2X,'SEQN',I4/2X, 1'HCLASS',I4,2X,'INITS',I4,2X,'INTN',I4,2X,'PR0B1',I4,2X,'PR0B2',I4,2X,'PR0B3',I4,2X,'PR0B4',I4/1)
105 FORMAT(5X,'INITS',I3,1)' I2,2X,'MOVET=I4,2X,'MOV3=I4
1,12,2X,'TYPE NUMBER'=I3/1)
123 FORMAT(5X,'SEQ=',I3,1)
124 FORMAT(1,1,2X,'TYPE NO.',14,2X,'INITIAL MOVES (MOV1=',I2,2X,'MOV2 1,12,2X,'MOV3=',I2,1)1)
END FILE 3
RETURN
END

SUBROUTINE CKTYP(CKGRP1, CKGRP2, NG0, FLAG)

THIS SUBROUTINE CHECKS A CATEGORY (GROUP) TO DETERMINE IF IS
CLOSE ENOUGH FOR A PATTERN MATCH. IT MUST NOT BE SEPARATED
BY MORE THAN ONE CATEGORY FOR TWO GROUPS IN A SEQUENCE.

IMPLICIT INTEGER (A-W)
DATA A1/A1/’A2’/A2/’A2’/
IF(CKGRP1.EQ.1.OR. CKGRP1.EQ.2.OR. CKGRP1.EQ.3) GOTO 256
IF(CKGRP1.EQ.1) GOTO 10
IF(CKGRP1.EQ.2) GOTO 20
WRITE(6,256)
256 FORMAT(5X,'CANT FIND A CHECK GROUP IN SUBROUTINE CKTYP IN CLSTY 1P SUBROUTINE //')
GOTO 700
5 VALUE=CKGRP1-2
IF (VALUE) 6, 7, 7
6 IF(CKGRP2.EQ.AI) GOTO 66
7 IF(FLAG)66, 665, 666
665 NG0=1
RETURN
666 NG0=0
RETURN
66 NG0=0
RETURN
7 IF(CKGRP2.EQ.A2) GOTO 66
IF (VALUE.EQ.0) GO TO 6
GOTO 72
10 IF(CKGRP2.EQ.1.OR. CKGRP2.EQ.2) GOTO 66
GOTO 76
20 IF(CKGRP2.EQ.2.OR.CKGRP2.EQ.3) GOTO 66
GOTO 76
760 RETURN
END

SUBROUTINE TOTMV(TOTAL,GROUP)
SUBROUTINE TO CLASSIFY COMBINATIONS OF ANY THREE MOVES TO
PRODUCE CATEGORIES OF GROUP 1,A1,2,A2,3.

IMPLICIT INTEGER (A-W)
COMMON/AVAIL4,AVAIL3,AVAIL2,NAM4(1000),NAM2(1000),NAM3(1000)
COMMON/TOP/BOTM,CATLG,CHAR,HIST,PLAYER,CATLEN,PATRNS,PATRNN
DIMENSION RLINK4(1000),LLINK4(1000),DOWN4(1000),RLINK3(1000),DOWN3(1000),DOWN2(1000)
EQUIVALENCE (NAM4(4),RLINK4(1),LLINK4(1),DOWN4(1)),(NAM2(2),DOWN12(1)),(NAM3(3),RLINK3(2),DOWN311)
DATA A1/A1',A1',A2'/A2'/A2'/A2'/A2'/
IF(TOTAL.EQ.3.OR.TOTAL.EQ.4) GOTO 4
IF(TOTAL.EQ.5) GOTO 5
IF(TOTAL.EQ.6) GOTO 6
IF(TOTAL.EQ.7) GOTO 7
IF(TOTAL.EQ.8.OR.TOTAL.EQ.9) GOTO 9
4 GROUP=1
RETURN
5 GROUP=A1
RETURN
6 GROUP=2
RETURN
7 GROUP=A2
RETURN
9 GROUP=3
RETURN
END

SUBROUTINE GETMV(NXTGRU,TMOVE2,PMOVE)
SUBROUTINE TO DETERMINE NEXT MOVE FROM PREDICTED GROUP.

IMPLICIT INTEGER (A-W)
COMMON/AVAIL4,AVAIL3,AVAIL2,NAM4(1000),NAM2(1000),NAM3(1000)
COMMON/TOP/BOTM,CATLG,CHAR,HIST,PLAYER,CATLEN,PATRNS,PATRNN
DIMENSION RLINK4(1000),LLINK4(1000),DOWN4(1000),RLINK3(1000),DOWN3(1000),DOWN2(1000)
EQUIVALENCE (NAM4(4),RLINK4(1),LLINK4(1),DOWN4(1)),(NAM2(2),DOWN12(1)),(NAM3(3),RLINK3(2),DOWN311)
DATA A1/A1/, A2/A2/

IF(NXTGRU.EQ.1) GOTO 10
IF(NXTGRU.EQ.A1) GOTO 11
IF(NXTGRU.EQ.2) GOTO 12
IF(NXTGRU.EQ.A2) GOTO 13
IF(NXTGRU.EQ.3) GOTO 14
IF(TMOVE2.EQ.2) GOTO 22
IF(TMOVE2.EQ.3) GOTO 23
IF(TMOVE2.EQ.4) GOTO 24

10 PMOVE=2
RETURN
3 PMOVE=1
RETURN
11 IF(TMOVE2.EQ.2) GOTO 22
IF(TMOVE2.EQ.3) GOTO 23
IF(TMOVE2.EQ.4) GOTO 24

22 PMOVE=3
RETURN
23 PMOVE=2
RETURN
24 PMOVE=1
RETURN
12 IF(TMOVE2.EQ.3) GOTO 33
IF(TMOVE2.EQ.4) GOTO 34
IF(TMOVE2.EQ.5) GOTO 35

33 PMOVE=3
RETURN
34 PMOVE=2
RETURN
35 PMOVE=1
RETURN
13 IF(TMOVE2.EQ.4) GOTO 44
IF(TMOVE2.EQ.5) GOTO 45
IF(TMOVE2.EQ.6) GOTO 46

44 PMOVE=3
RETURN
45 PMOVE=2
RETURN
46 PMOVE=1
RETURN
14 IF(TMOVE2.EQ.6) GOTO 56
IF(TMOVE2.EQ.5) GOTO 55

56 PMOVE=2
RETURN
55 PMOVE=3
RETURN
END
SUBROUTINE STRTOT(STRT)

SUBROUTINE TO DETERMINE WHERE THE LAST MOVE FITS IN THE TOTAL
OF TYPES OF MOVES IN THE TEMPORARY MEMORY HEADER AND UPDATE THE
APPROPRIATE CELLS IN THE HEADER NODES.

IMPLICIT INTEGER (A-W)
COMMON/TAC/VALUE(7,3),RTABN(3,3),RTABS(3,3),ATABN(3),ATABS(3),
1SN,SS,WN,WS,RS,RW,NC,INRAND,YRAND,FFLAG,CPN,CP8,STRATN,STRATS,
2PNLD,IGN,IS,S,ECXN,ECNXS,PN,PS,MFACTN,MFACTS
COMMON/TV/AVAIL2,AVALI3,AVALI2,NAME4(1000),NAME2(1000),NAME3(1000)
COMMON/TC/TDF,BOON,CATLG,CHAR,HIST,PLAYER,CATLGN,PATRNS,PATRNN
DIMENSION RLKN4(1000),DLN4(1000),RLKN3(1000),DOWN3
1(1000),DOWN2(1000)
EQUIVALENCE (NAME4(4),RLNK4(3),LLNK4(2),DOWN4(1)),(NAME2(2),DOWN
12(1)),(NAME3(3),RLNK3(2),DOWN3(1))

IF (STRAT=9) GOTO 100
STRT=LLNK4(PATRNS)
VALU1=STRATN-2
IF (VALU1) 10,20,30
10 NAME4(STRT)=NAME4(STRT)+1
RETURN
20 RLKN4(STRT)=RLKN4(STRT)+1
RETURN
30 LLLK4(STRT)=LLNK4(STRT)+1
RETURN
100 STRT=LLNK4(PATRNS)
VALU1=STFATS-2
IF (VALU1) 11,21,31
11 NAME4(STRT)=NAME4(STRT)+1
RETURN
21 RLNK4(STRT)=RLNK4(STRT)+1
RETURN
31 LLLK4(STRT)=LLNK4(STRT)+1
RETURN
END

SUBROUTINE TOTLM((MOVE,TOTDF,BYPASS,INCNT,MCNT,TOTAL,TOTAL2)

THIS SUBROUTINE TOTALS THE THREE MOVES IN TEMPORARY MEMORY
FOR BOTH PATRNN AND PATRNS.

IMPLICIT INTEGER (A-W)
COMMON/TAC/VALUE(7,3),RTABN(3,3),RTABS(3,3),ATABN(3),ATABS(3),
1SN,SS,WN,WS,RS,RW,NC,INRAND,YRAND,FFLAG,CPN,CP8,STRATN,STRATS,
2PNLD,IGN,IS,S,ECXN,ECNXS,PN,PS,MFACTN,MFACTS
COMMON/TV/AVAIL2,AVALI3,AVALI2,NAME4(1000),NAME2(1000),NAME3(1000)
COMMON/TC/TDF,BOON,CATLG,CHAR,HIST,PLAYER,CATLGN,PATRNS,PATRNN
COMMON/TDF/MOVES,MIVEN
COMMON/TDF3/TYPE(20),INIT(20)
DIMENSION RLINK4(1000), LLINK4(1000), DOWN4(1000), RLINK3(1000), DOWN3(1000),
DOWN2(1000)
EQUIVALENCE (NAME4(4), RLINK4(3), LLINK4(2), DOWN4(1)), (NAME2(2), DOWN2(1))
DATA A1,'A1',A2,'A2'/
MCNT=1
TOTAL2=0
TOTAL=0
IF(MOVE.EQ.MOVE)GOTO 100
TOPMN=MOVE
106 TOTAL=TOTAL+NAME2(TOPMN)
IF(DOWN2(TOPMN),EQ.0) GOTO 107
TOPMN=DOWN2(TOPMN)
TOTAL2=TOTAL2+NAME2(TOPMN)
GOTO 106
100 TOPMS=MOVE
108 TOTAL=TOTAL+NAME2(TOPMS)
IF(DOWN2(TOPMS),EQ.0) GOTO 107
TOPMS=DOWN2(TOPMS)
TOTAL2=TOTAL2+NAME2(TOPMS)
GOTO 108
107 IF(BYPASS.EQ.0) GOTO 110

BYPASS IF YOU DO NOT WANT TO TOTAL MOVES IN THE INITIAL PATTERN.

62 IF(MCNT.GT.INCNT) GOTO 64
TOPI=INIT(MCNT)
ITOTAL=0
ITOTAL=ITOTAL+NAME4(TOPI)
ITOTAL=ITOTAL+RLINK4(TOPI)
ITOTAL=ITOTAL+LLINK4(TOPI)
IF(TOTAL.EQ.ITOTAL) GOTO 65
MCNT=MCNT+1
GOTO 62
65 TRUE=1
RETURN
64 TRUE=0

"0" => NOT TRUE NO MATCH IN THE INITIAL LIBRARY.
"1" => TRUE THERE IS A MATCH IN INITIAL LIBRARY.
110 RETURN
END
SUBROUTINE MOVEPL
SUBROUTINE TO PUT OLD PLAYER TO THE TOP OF THE CATALOGUE.

IMPLICIT INTEGER (A-H)
COMMON/T/AVAIL4,AVAIL3,AVAIL2,NAME4(1000),NAME2(1000),NAME3(1000)
COMMON/ID/TOP,ROTM,CATLG,HIST,PLAYER,CATLGN,PATRNS,PATPN
DIMENSION RLINK4(1000),LLNK4(1000),DOWN4(1000),RLINK3(1000),DOWN3
1(1000),DOWN2(1000)
EQUIVALENCE (NAME4(4),RLINK4(3),LLNK4(2),DOWN4(1)),(NAME2(2),DOWN
12(1)),(NAME3(3),RLINK3(2),DOWN3(1))
MV=CATLG
1 IF(MV.EQ.TOP) GOTO 4
2 IF(DOWN3(MV).EQ.TOP) GOTO 2
3 MV=DOWN3(MV)
GOTO 1
4 MV=CATLG
RETURN
END

SUBROUTINE SEARCH(PLY)
SUBROUTINE TO FIND OUT IF A PLAYER IS IN THE CATALOGUE
OF PLAYERS

IMPLICIT INTEGER (A-H)
COMMON/T/AVAIL4,AVAIL3,AVAIL2,NAME4(1000),NAME2(1000),NAME3(1000)
COMMON/ID/TOP,ROTM,CATLG,HIST,PLAYER,CATLGN,PATRNS,PATPN
DIMENSION RLINK4(1000),LLNK4(1000),DOWN4(1000),RLINK3(1000),DOWN3
1(1000),DOWN2(1000)
EQUIVALENCE (NAME4(4),RLINK4(3),LLNK4(2),DOWN4(1)),(NAME2(2),DOWN
12(1)),(NAME3(3),RLINK3(2),DOWN3(1))
TOP=CATLG
2 IF(NAME3(TOP).EQ.PLAYER) GOTO 1
3 IF(DOWN3(TOP).EQ.0) GOTO 3
4 TOP=DOWN3(TOP)
GOTO 2
5 PLY=0
RETURN
6 PLY=1
RETURN
END
SUBROUTINE CLSMVS(MOV,INCNT,PCNT,PCNTM,INCNTR)

SUBROUTINE TO CLASSIFY INITIAL MOVES FROM SOUTHS PATTERN.

IMPLICIT INTEGER (A-W)
COMMON/TAVAIL2,AVAIL3,AVAIL2,NAME4(1000),NAME2(1000),NAME3(1000)
COMMON/TOP5/TYPENU,TYPE1,TYSEQ,TYNOR
COMMON/TO/TP2,BATM,CHATL,PLAYER,CATLGN,PATRNS,PATPNN
COMMON/TO/SEQ(20),SCCNTR
COMMON/TO/TYPE(20),INT(20)
COMMON/TO/STK(10),STKN(10),STKNM(10)
COMMON/CLSTREFF,REFF,OFFEN,BOFFEN,PMOVES,PMOVES
COMMON/JEC/VALUE(7,3),RTABN(3,3),RTABS(3,3),ATABN(3),ATABS(3)
SN,SS,WN,RS,WR,NC,INRAND,INAND,FFLAG,CNP,CPN,STRATN,STRATS,
2PNLD,ISN,ISS,ECNXS,PN,PS,FACTXN,FACXNS
COMMON/TO/P/GRPS,GRPUN,TMOVES,TMOVEN,MCNT,CONTR
COMMON/TO/P/RTHIST,NTHIST
DIMENSION RLINK4(1000),RLINK3(1000),DOWN4(1000),DOWN3
1(1000),DOWN2(1000)
EQUIVALENCE (NAME2(4),RLINK4(3),RLINK4(2),DOWN4(1)),(NAME2(2),DOWN
12(1)),(NAME3(3),RLINK3(2),DOWN3(1))

MOV2=NC
CCNT=1
DCNT=1
MCNT=1
MOVES=DOWN4(PATRNS)
IF INCNT .EQ. 0 GOTO 82
9
TO/MP=Moves
IF(MOV2.GT.3) GOTO 35
10
IF(MCNT.GT.INCNT) GOTO 20
TOPI=INIT(MCNT)
IF(NAME4(TOPI).EQ.NAME2(TOMP2)) GOTO 30
MCNT=MCNT+1
GOTO 10
20
IF(PCNT.GT.0) GOTO 21
DEFEN=0
"0H=> NO DEFENSIVE STRATEGY IS AVAILABLE. FIND BEST DEFENSE
FOR MIXED MOVE MATCH STRATEGY.
MCNT=1
43
TO/MP=Moves
41
IF(MCNT.GT.INCNT) GOTO 40
TOPI=INIT(MCNT)
IF(RLINK4(TOPI).EQ.NAME2(TOMP2)) GOTO 45
MCNT=MCNT+1
GOTO 41
40 IF(PCNTM.GT.0) GOTO 42
PDEF=0
PMOVES=0
C
"""" " " " BEST DEFENSE FROM MIXED MOVE STRATEGY.
RETURN
45 DCNT=DCNT+1
TOPM=DCWN2(TOPM)
IF(NAME4(TOPI).EQ.NAME2(TOPM)) GOTO 40
MCNT=MCNT+1
DCNT=1
GOTO 43
49 PNTR=MCNT
PCNTM=PCNTM+1
STKM(PCNTM)=PNTR
PTRM=POINTER
MCNT=MCNT+1
DCNT=1
GOTO 43
42 IF(PCNTM.GT.1) GOTO 47
TOPI=INIT(PTRM)
MCNT=PNTR
46 PMOVES=LLINK4(TOPI)
NORS=0
CALL CLSTYP(NORS)
PDEF=DEFES
DEFES=0
RETURN
47 PTRM=STKM(PCNTM)
PCNTM=PCNTM+1
TOPI=INIT(PTRM)
MCNT=PNTR
GOTO 46
30 DCNT=DCNT+1
TOPM=DCWN2(TOPM)
IF(RLINK4(TOPI).EQ.NAME2(TOPM)) GOTO 31
DCNT=1
MCNT=MCNT+1
GOTO 9
31 PNTR=MCNT
PCNT=PCNT+1
STK(PCNT)=PNTR
PTR=POINTER
DCNT=1
MCNT=MCNT+1
GOTO 9
21 IF(PCNT.GT.1) GOTO 25
TOPI=INIT(PTR)
MCNT=PTR
24 MOVES=LLINK4(TOPI)
NORSO=0
CALL CLSTYP(NORSO)
RETURN
25 PTR=STK(PCNT)
TOPI=INIT(PTR)
MCNT=PTR
GOTO 24

C GET STRATEGY FOR THE FORTH MOVE AND SET UP ADDITIONAL INITIAL MOVES CELLS.
C
35 IF(PCNT.EQ.0) GOTO 36
PTR=STK(PCNT)
PCNT=PCNT-1
TOPMS=MOVES
DCNT=1
39 IF(PCNT.EQ.3) GOTO 38
TOPMS=DOWN2(TOPMS)
DCNT=DCNT+1
GOTO 35
38 TOPI=INIT(PTR)
MCNT=PTR
IF(LLINK4(TOPI).EQ.NAME2(TOPMS)) GOTO 55
GOTO 35
60 MOVE=1
MOVE=MOVES
CKINCT=INCNT
CALL INTLIB(INCNT,MOVE,TOPI,INCNTR,MV)
C NOW CELL HAS BEEN ADDED
MOVES=DOWN4(PATRNS)
TOPS=MOVES
NAME4(TOPI)=NAME2(TOPMS)
TOPMS=DOWN2(TOPMS)
RLINK4(TOPI)=NAME2(TOPMS)
TOPMS=DOWN2(TOPMS)
LLINK4(TOPI)=NAME2(TOPMS)
DOWN4(TOPI)=0
GOTO 80
36 DEFES=0
MOVE=MOVES
BYPASS=1
CALL TOTALM(MOVE,TOTRUE,BYPASS,INCNT,MCNT,TOTAL,TOTAL2)
IF(TOTRUE)64,64,65
65 NORSO=0
IF(DOWN4(MCNT).EQ.0) GOTO 650
TOPI=INIT(MCNT)
IF(DOWN4(TOPI).EQ.0) GO TO 650
RETURN
82 IF(MOV2.GT.3)GOTO 60
DEFES=0
PDEFES=0
PMOVES=0
RETURN

SUBROUTINE CLSTYP(NORSO)
SUBROUTINE TO DETERMINE IF A PLAYER FITS A PARTICULAR
PATTERN IN THE TERM LEXICARY AND IF SO OBTAIN THE PREDICTED
MOVE FOR THE BEST DEFENSIVE OR OFFENSIVE MOVE FOR THE
MATCHED PATTERN.

NORSO===> NORTH OR SOUTH
1===> NORTH (COMPUTER)
0===> SOUTH (PLAYER)

IMPLICIT INTEGER (A-W)
COMMON/T/AVAL14,AVAIL3,AVAIL2,NAME4(1000),NAME2(1000),NAME3(1000)
COMMON/T/DD/TOP,ROTM,CATLN,CHAR,HIST,PLAYER,CATGN,PATRNS,PATRN
COMMON/T/DS/TYPE1,TYPES,TYSO,TYMD
COMMON/T/DP/CLASS,CLASSN,TMOVES,TMOVES,MCNT,CONTR
COMMON/T/DEP/SEQ(20),SCONTR
COMMON/T/DP/TYPE2(20),INT(20)
COMMON/CL/CLASS,CLASSN,PDEFES,PMOVES,PMOVES
COMMON/JEC/VICLUE(7,3),RTARN(3,3),RTARS(3,3),ATARN(3),ATARS(3),
1SN,SS,WN,WS,RS,PW,NC,HAND,YAND,EFLAG,CPN,CPS,STRATN,STPATS,
2PNOLT,ISN,ISS,PCNXN,PCNXS,PN,PS,FACTNX,FACTXS
DIMENSION RLINK4(1000),LLINK4(1000),DOWN4(1000),RLINK3(1000),DOWN3
1(1000),DOWN2(1000)
EQUIVALENCE (NAME4(4),RLINK4(3),LLINK4(2),DOWN4(1)),(NAME2(2),DOWN
12(1)),(NAME3(3),RLINK3(2),DOWN3(1))
TOT = 0
TFLAG = 0
IF (NC .GT. 3) GOTO 11
CALL FINDTY(MCNT, TYPENU, NORSO)
I = (TYPENU .EQ. 0) GOTO 100
TYPE1 = TYPENU
IF (NC .EQ. 3) GOTO 10
IF (NC .EQ. 2) GOTO 1
IF (NORSO .EQ. 1) GOTO 2
TYSOU = TYPE1
PMOVES = NAME3(TYPE1)
RETURN
2 PMOVEN = NAME3(TYPE1)
TYNOR = TYPE1
RETURN
1 IF (NORSO .EQ. 1) GOTO 3
PMOVES = RLINK3(TYPE1)
TYSOU = TYPE1
RETURN
3 PMOVEN = RLINK3(TYPE1)
TYNOP = TYPE1
RETURN
10 MOVE = DOWN4(PATRNS)
IF (NORSO .EQ. 1) MOVE = DOWN4(PATRNN)
6 TOT = TOT + NAME2(MOVE)
IF (DOWN2(MOVE) .EQ. 0) GOTO 5
MOVE = DOWN2(MOVE)
GOTO 6
5 IF (NORSO) 555, 554, 555
554 TYSOU = TYPE1
GOTO 556
555 TYNOR = TYPE1
556 FINDGU = DOWN3(TYPE1)
NXTGRU = NAME2(DOWN2(FINDGU))
TMOVE2 = TOT
CALL GETMVNXGRU, TMOVE2, PMOVE
IF (NORSO .EQ. 1) GOTO 50
PMOVES = PMOVE
DEFFS = NAME3(TYPE1)
OFFFS = RLINK3(TYPE1)
TYPE1 = 0
RETURN
50 PMOVEN = PMOVE
DEFEN = NAME3(TYPE1)
OFFEN = RLINK3(TYPE1)
IF (DEFES .EQ. 0) GOTO 90
IF (DEFEN .EQ. 0) GOTO 90
IF (OFFES .EQ. 0) GOTO 51
IF(NC.LE.3) RETURN
GOTO 11
51 IF (NC.GT.4 .AND. TFLAG.EQ.0) RETURN
TFLAG=0
FINDGU=DOWN3(DEFEN)
TYNOR=DEFEN
NXTRU=NAME2(DOWN2(FINDGU))
CALL GETV((NXTRL,TMOVE2,PMOVE)
PMOVE=PMOVE
TYPE1=0
RETURN
11 IF(TYPENU.EQ.0) GOTO 100
IF(NORSO)777,770,777
770 IF(TYSOU.NE.0) GOTO 110
TYSOU=TYPENU
110 TYPE1=TYSOU
GOTO 112
777 IF(TYNOR.NE.0) GOTO 111
TYNOR=TYPENU
111 TYPE1=TYNOR
112 DOWNTY=DOWN3(TYPE1)
FINDGU=DOWN2(DOWNTY)
IF(NORSO.EQ.1) GOTO 41
PATRN= PATRN
TMOVE2=TMOVE
40 STRP= 41 LINK4(PATRN)
TOPG=DOWN4(STRP)
DOWNPT=TOPG
GOTO 42
41 PATRN= PATRN
TMOVE2=TMOVE
GOTO 40
42 IF(DOWN2(DOWNPT).EQ.0) GOTO 45
IF(DOWN2(FINDGU).EQ.0) GOTO 48
TOPGU=DOWNPT
TOPTY=DOWNTY
DOWNPT=DOWN2(DOWNPT)
DOWNTY=DOWN2(DOWNTY)
FINDGU=DOWN2(DOWNTY)
GOTO 42
45 IF(NAME2(DOWNPT).EQ.NAME2(DOWNTY)) GO TO 47
FLAG=0
IF(NAME2(TOPGU).EQ.NAME2(TOPTY)) GOTO 48
CKGRP1=NAME2(TOPGU)
CKGRP2=NAME2(TOPTY)
80 CALL CKTP(KGRP1,KGRP2,NOGO,FLAG)
IF(NOGO.EQ.1) GOTO 49
47 NXTRU=NAME2(FINDGU)
CALL GETMV(NXTGRU,TMOVE2,PMOVE);
IF(NORSQ.EQ.1) GOTO 43
PMOVES=PMOVE
DEFFS=NAME3(TYPE1)
DEFFS=RLINK3(TYPE1)
TYPE1=0
RETURN
43 PMOVEN=PMOVE
DEFFEN=NAME3(TYPE1)
DEFFEN=RLINK3(TYPE1)
IF(DEFFS.EQ.DEFFEN) GO TO 90
IF(DEFFEN.EQ.0) GO TO 90
IF(0FFSES.EQ.DEFFEN) GO TO 51
RETURN
48 FLAG=1
CKGRP1=NAME2(DOWNPT)
CKGRP2=NAME2(DOWNTY)
GOTO 80
49 CONT=1
IF(CONT.GT.CONT) GOTO 56
TOPGP=DOWN3(TYPE(CONT))
NXTGP=DOWN2(TOPGP)
IF(NAME2(TOPGP),EO,NAME2(TOPGP)) GOTO 57
CKGRP1=NAME2(TOPGP)
CKGRP2=NAME2(TOPGP)
CALL CKTYPE(CKGRP1,CKGRP2,NOGO,FLAG)
IF(NOGO.EQ.0) GOTO 57
CONT=CONT+1
GOTO 53
57 FLAG=1
IF(NAME2(DOWNPT),EO,NAME2(NXTGP)) GOTO 58
CKGRP1=NAME2(DOWNPT)
CKGRP2=NAME2(NXTGP)
CALL CKTYPE(CKGRP1,CKGRP2,NOGO,FLAG)
IF(NOGO.EQ.0) GOTO 58
CONT=CONT+1
GOTO 53
56 IF(NORSQ.EQ.1) GOTO 52
PMOVES=0
RETURN
52 PMOVEN=0
RETURN
58 IF(NORSQ.EQ.1) GOTO 55
TYPE1=TYPE(CONT)
NXTGRU=NAME2(NXTGP)
TYPE1=TYPE(CONT)
GO TO 471
55 TYNOR=TYPE(CONT)
TYPE = TYPE(CONT)
NXTGRU = NAME2(NXTGP)
TFLAG = 1
GO TO 671
100 PMOVES = 0
PMOVED = 0
RETURN
90 TYPE1 = 0
RETURN
END

SUBROUTINE CLSMVNI(MOV, INCNT, PCNTN, PCNTNM, INCNTP)

SUBROUTINE TO CLASSIFY INITIAL MOVES FROM NORTH'S PATTERN.

IMPLIED INTEGER (A-W)
COMMON/TA/AVAIL, AVAIL1, AVAIL2, NAME4(1000), NAME2(1000), NAME3(1000)
COMMON/TDA/TO, ROTM, CATLG, CHAR, HI, PLAY, CATLP, PATRN, PATRN
COMMON/TDP3/TYP(20), INIT(20)
COMMON/TDP4/SEQ(20), SCNTR
COMMON/TDP5/STK(10), STKN(10), STKNM(10)
COMMON/TDP9/TYPE1, TYSOU, TYKND
COMMON/TDP10/MHIST, NAHIST
COMMON/JEC/VALUE(7, 3), RTABM(3, 3), RTABT(3, 3), ATABN(3), ATABT(3),
1SN, SS, WN, WS, RS, PW, NC, IRRAND, YRAND, FFLAG, CNP, CPS, STRATN, STRATS,
2PNL, TSN, TS, ECNXN, ECNXS, PN, PC, FACTN, FACTS
COMMON/TDP7/CONT, GROUPN, TMOVFS, TMOV 
DIMENSION RLINK4(1000), LLINK4(1000), DOWN4(1000), RLINK3(1000), DOWN3
1(1000), DOWN2(1000)
FOU!VALHCCE (NAME4(4), RLINK4(3), LLINK4(2), DOWN4(1)), (NAME2(2), DOWN
12(1)), (NAME3(3), RLINK3(2), DOWN3(1))
MOV2 = NC
DCNT = 1
DCNT = 1
DFTM = 1
MCNT = 1
PMOV = DOWN4(PATRN)
IF(INCNT EQ 0) GO TO 82
9 TDP0 = MOV1
IF(MOV2 GT 3) GO TO 35
10 IF(MCN0 = GT INCNT) GO TO 20
TOP1 = INIT(MCNT)
IF(NAME4(TOP1), EQ, NAME2(TOPMN)) GO TO 30
MCNT = MCNT + 1
GO TO 10
20 IF(PCNTN GT 0) GO TO 21
OFFEN=0
"OM=>' NO DEFENSIVE STRATEGY IS AVAILABLE. FIND BEST DEFENSE
FOR MIXED MOVE MATCH STRATEGY.
MCNT=1
43 TOPMN=M0VEN
41 IF(MCNT.GT.INCNT)GOTO 40
TOPI=INIT(MCNT)
IF(PLINK4(TOPI).EQ.NAME2(TOPMN))GOTO 45
MCNT=MCNT+1
GOTO 41
40 IF(PCNTNM.GT.0)GOTO 42
OFFEN=0
PMOVEN=0
"OM=>' BEST DEFENSE FROM MIXED MOVE STRATEGY.
RETURN
45 DCTM=DCTM+1
TOPMN=DOWM2(TOPMN)
IF(NAME4(TOPI).EQ.NAME2(TOPMN))GOTO 49
MCNT=MCNT+1
DCTM=1
GOTO 42
49 POINTR=MCNT
PCNTNM=PCNTNM+1
STKNM(PCNTNM)=POINTR
PTRM=POINTR
MCNT=MCNT+1
DCTM=1
GOTO 43
42 IF(PCNTNM.GT.1)GOTO 47
TOPI=INIT(PTRM)
MCNT=PTRM
46 PMOVEN=LLINK4(TOPI)
NORSE=1
CALL CLSTYP(NORSE)
OFFEN=OFFEN
OFFEN=0
RETURN
47 PTRM=STKNM(PCNTNM)
PCNTNM=PCNTNM-1
TOPI=INIT(PTRM)
MCNT=PTRM
GOTO 46
30 DCT=DCT+1
TOPMN=DOWM2(TOPMN)
IF(PLINK4(TOPI).EQ.NAME2(TOPMN))GOTO 31
DCT=1
MCNT=MCNT+1
GOTO 9
31    POINTR=MCNT
PCNTN=PCNTN+1
STKM(PCNTN)=POINTR
PCTR=POINTR
DCNT=1
MCNT=MCNT+1
GOTO 0
21    IF(PCNTN.GT.1)GOTO 25
TOPI=INIT(PTR)
MCNT=PTR
24    PMOVEN=LLINK4(TOPI)
NORSO=1
CALL CLSTYP(NORSO)
RETURN
25    PTR=STKN(PCNTN)
TOPI=INIT(PTR)
MCNT=PTR
GOTO 24
GET STRATEGY FOR THE FOURTH MOVE AND SET UP ADDITIONAL INITIAL MOVEN CELLS.
35    IF(PCNTN.EQ.0) GOTO 36
PCTR=STKN(PCNTN)
PCNTN=PCNTN-1
TOPMN=PCEVEN
DCNT=1
39    IF(DCNT.EQ.3) GOTO 38
TOPMN=DOWN2(TOPMN)
DCNT=DCNT+1
GOTO 36
38    TOPI=INIT(PTR)
MCNT=PTR
IF(LLLKNK4(TOPI),EQ.,NAME2(TOPMN)) GOTO 55
GOTO 35
60    MOV1=1
MOVE=MVEN
CHKINCT=INCNT
CALL INTLIB(INCNT,MOVE,TOPI,INCNR,MOV)
IF(CHKINCT,EQ.,INCNT) GOTO 81
NOW CELL HAS BEEN ADDED
MOVEN=DOWN4(PATRNN)
TOPMN=MVEN
NAME4(TOPI)=NAME2(TOPMN)
TOPMN=DCWN2(TOPMN)
RLLINK4(TOPI)=NAME2(TOPMN)
TOPMN=DCWN2(TOPMN)
LLLKNK4(TOPI)=NAME2(TOPMN)
DOWN4(TOPI)=0
GOTO 60
OFFEN=0
BYPASS=1
MOVE=MOVEN
CALL TOTALM(MOVE,TOTTRUE,BYPASS,INCNT,MCNT,TOTAL,TOTAL2)
IF(TOTTRUE)64,64,65
NORSO=1
IF(MCNT .GT. INCNT) GO TO 650
I = INIT(INCNT)
IF(DOW4(TOP1),EQ.0) GO TO 650
TYPEN= DOW4(TOP1)
650 CALL CLSTYP(NORSO)
OFFEN=OFFEN
OFFEN=0
GOTO 60
64 OFFEN=0
OFFEN=0
GOTO 60
55 NORSO=1
CALL CLSTYP(NORSO)
GOTO 60
80 RLHIST=INIT(INCNT)
RETURN
91 RLHIST=TOP1
RETURN
82 IF(MOV2.GT.3)GOTO 60
OFFEN=0
OFFEN=0
PMOVEN=0
RETURN
END

SUBROUTINE FINDTY(MCNT,TYPEN,NORSO)
SUBROUTINE TO FIND A HEADER NODE IN THE TYPE LIBRARY IF A
PLAYER HAS PLAYED BEFORE AND IS IN THE CATALOGUE OF PLAYERS.

IMPLICIT INTEGER(A-W)
COMMON/T/AVAIL4,AVAIL3,AVAIL2,NAME4(1000),NAME2(1000),NAME3(1000)
COMMON/TOT/TOP,BOTM,CATLG,CHAR,HIST,PLAYER,CATLGN,PATRNS,PATRNN
COMMON/TDP4/SEQ(20),SCCTR
COMMON/TDP3/TYP3(20),INIT(20)
COMMON/JEC(VALUE(7,3),RTABN3(3,3),RTARS3(3,3),ATABN3),ATARS3,
ISP,SS,WS,RW,NC,IRAN,VRAND,RF,CPN,CPS,STRATN,STRATS,
2PNOLD,ISP,SS,ECNXN,EcnxS,PN,PS,FACTN,FACTXS
DIMENSION RLINK4(1000),LLINK4(1000),DOW4(1000),RLINK3(1000),DOWN3
1(1000),DOWN2(1000)
EQUIVALENCE (NAME4(4),RLINK4(3),LLINK4(2),DOWN4(1)), (NAME2(2), DOWN
1(1)), (NAME3(3),RLINK3(2),DOWN3(1))
IF (NC.LT.2) GOTO 20
IF (NC.GT.3) GOTO 20
TOP=INITMCT)
IF (DOWN4(TOPI).EQ.0) GOTO 20
TYPE=DOWN4(TOPI)
RETURN
20 IF (TOP.EQ.1) GOTO 21
IF (NAME4(HIST).EQ.0) GOTO 22
INIT=NAME4(HIST)
IF (DOWN4(INITN).EQ.0) GOTO 22
TYPE=DOWN4(INITN)
RETURN
21 IF (RLINK4(HIST).EQ.0) GOTO 32
INIT=RLINK4(HIST)
IF (DOWN4(INITN).EQ.0) GOTO 32
TYPE=DOWN4(INITN)
RETURN
22 IF (LLINK4(HIST).EQ.0) GOTO 42
SEON=LLINK4(HIST)
RTOP=SEON
40 IF (DOWN3(RTOP).EQ.0) GOTO 43
RTOP=DOWN3(RTOP)
GOTO 40
43 TYPE=NAME3(RTOP)
RETURN
32 IF (DOWN4(HIST).EQ.0) GOTO 42
SEON=RLINK4(HIST)
RTOP=SEON
GOTO 40
42 TYPE=0
RETURN
END
BIBLIOGRAPHY


15. Taber, C. W., Taber's Cyclopedic Medical Dictionary, Davis, 1965.

160


The purpose of this thesis is a discussion of developing human-like behavior in the computer. A theory of the human learning processes is first described. This leads to the presentation of a computer game which simulates the human capabilities of reasoning and learning. The program is required to make intelligent decisions based on past experiences and critical analysis of the present situation.
<table>
<thead>
<tr>
<th>KEY WORDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Artificial Intelligence</td>
</tr>
<tr>
<td>Brain Model</td>
</tr>
<tr>
<td>Computer Game Playing</td>
</tr>
<tr>
<td>Information Structures</td>
</tr>
<tr>
<td>Management Decision Simulation</td>
</tr>
<tr>
<td>Thinking Machine</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LINK A</th>
<th>LINK B</th>
<th>LINK C</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROLE</td>
<td>WT</td>
<td>ROLE</td>
</tr>
</tbody>
</table>

Artificial Intelligence
Brain Model
Computer Game Playing
Information Structures
Management Decision Simulation
Thinking Machine