



# DECUS

## PROGRAM LIBRARY

DECUS NO.	FOCAL8-149
TITLE	CHECKERS
AUTHOR	Paul M. Klinkman
COMPANY	North Smithfield Jr. Sr. High School Woonsocket, Rhode Island
DATE	April 1, 1971
SOURCE LANGUAGE	FOCAL



## CHECKERS

DECUS Program Library Write-up

DECUS No. FOCAL8-149

### ABSTRACT

The object of this program is for the player to beat the computer in a game of checkers.

The checker board consists of 8 rows of 8 spaces, stored in the computer as A(1)-A(64). The checkers ( A(1)=1 is a computer's checker on space 1, A(1)=-1 is a player's checker) are to be moved or jumped forward diagonally until they reach the last row; they are then made kings. The computer's checkers are made queens to distinguish them from the player's kings. (A(1)=2 is a computer's queen, A(1)=-2 is a player's king.) The checkers never land on 32 of the spaces; therefore, group 8 is used to find if a space N is used. This saves the space of 32 variables. Because of the strange nature of the board, checkers can't go off one side of the board to the other side.

### USAGE

The computer first checks the contents of each space for its move. If a space has a computer's checker in it, two possible moves are recorded: M(D) and M(D+1). The space being moved from is stored in the thousands and hundreds place, while the space being moved to (the original space + 7 or 9) is stored in the tens and ones place. Four possible moves are recorded for queens.

Next the computer throws out all illegal moves by setting all such moves M(T)=0. At the same time it checks for jumps and, should it find one, executes it.

Finding no jumps, the computer checks to see if it can "thrust" a checker forward. Said checker would have a checker behind it. If the first checker gets jumped, the opposition's checker gets jumped too, and the computer with its handicap comes out ahead. This is why playing that a person has to jump is preferable; the program was constructed with this rule in mind. Also, the back four checkers, having nothing behind them, aren't moved until necessary.

If "thrusts" are found, one of them is randomly chosen and executed. If not, one of all the legal moves is chosen and executed. This program had to be altered radically to fit and there is almost no room for refinements in the computer's technique, improvements in input and output, or acknowledgements of victory or defeat. FOCAL, 1969, which this program was written for, has a strange random number generator. Should a different version of FOCAL be used where it gives out random numbers between -1 and 1, use

```
06.10 s T1=1+FABS(FITR(FRAN( )*(T-1) ) )
```

Next the program prints the checkerboard using the letters C, O, Q, K, and using # for an empty space.

The player's move is then asked for. The player must find the space of the checker he wants to move and the space he wants to move to on the board first typed out with comments at the start of the program. The computer figures out if the move is a jump by seeing if both numbers are odd or both even. Then it is the computer's move.

The computer can't test the legality of the player's moves. However, most illegal moves either erase the player's checker or add variables to fill up memory.

Only Ctrl-C will stop the program. Another game may be started by erasing the variables and feeding part III of the tape in.