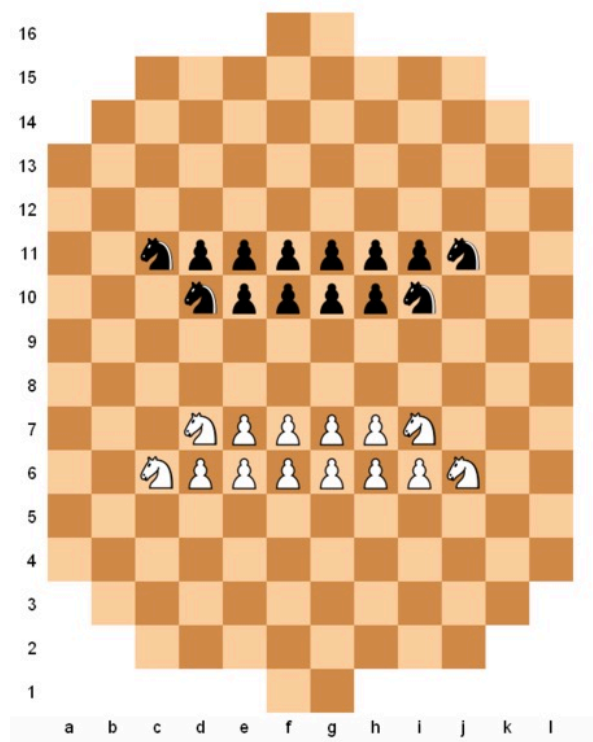


AI - Camelot

Introduction

You must have played chess, right? It's an all time favorite strategy game. Well this one is quite a similar one, but of course with a different strategy.

You have a board of 160 squares, which is roughly rectangular (12×14), with three squares removed from each of the four corners, and four extra squares extending outside the main rectangle, two each at the top and bottom of the board. These two-square areas are called the *castles*. Each player starts the game with fourteen pieces: four *knight*s and ten *men*, set up as shown (see diagram).



Problem Statement

The objective of the game is to be the first player to occupy the opponent's castle with two of your own pieces, or to capture *all* of your opponent's pieces while retaining two or more of your own pieces.

Event Structure

The event will be conducted in one round.

Each bot will be matched against every other bot 6 times. In a match, the two competing bots will play the game several times. The bot who wins maximum number of games will win the match.

Event Rules

- Both knights and men can move either horizontally, vertically, or diagonally in three ways, as follows:

- One space in any direction (like a **king** in chess). This is called a *plain move*.
 - A leaping move (called *cantering*) over an adjacent friendly piece to a vacant space immediately beyond. Multiple leaps over a player's own pieces are permitted. Canter is always optional (never obligatory).
 - A jumping move over an adjacent enemy piece to a vacant space immediately beyond. The enemy piece is captured and removed from the board. As in checkers, multiple jumps are allowed, and capturing is obligatory whenever it is possible.
 - Men may make any of the three moves, but only one type of move per turn. Knights have a fourth option: a combination move consisting of a *canter* immediately followed by a *jump* (capture). This is called the *knight's charge*. The knight may, in this single move, perform multiple canters (or just one), followed by multiple jumps (or just one); but the canter(s) must precede the jump(s). A knight may not combine a plain move with a canter or a jump.
- Your bot should not overwrite any opponent's piece's position. It should only overwrite the position of one of its pieces.
 - Your bot should make legal moves only.
 - Your bot should read the board file and print the previous position and the next position of your piece generated in a text file – 'output.txt'. (Eg : if your knight moves from C4 to C5, the generated output.txt file should have – "C4 C5" in the first line)
 - If it rewrites any of the other piece's position, it will be disqualified.

The organizers reserve all rights to change any or all of the above rules.

Specifications

- Your bot will be provided with a board file.
- The first line of the board file describes the team with which you will play (0 for black and 1 for white).
- The next 160 lines will describe the board (starting from the top two squares as shown in the diagram above). Each line will contain two words NOT separated by a space. The first word will be the position (refer the shown diagram) and the second word have the information about the piece at that position.
- For second word, first letter will be the team (B-Black and W-White) and second letter will be the piece (K-Knight and M-Men). In case of an unoccupied square, 'U' will be there.
- Eg (refer the diagram above) : "F16BK" denotes that the square of F-16 will have a Black Knight. Similarly "C09U" denotes that the square of C-9 in an unoccupied one.
- FYI, this is the sample board file - <http://pastebin.com/qRzvrVNM>

Contacts

Abhishek Sah
7357277925
sah.1@iitj.ac.in

Vishesh Mistry
8094696984
mistry.1@iitj.ac.in