AI CHALLENGE

Introduction:

Connect Four (also known as **Captain's Mistress**, **Four Up**, **Plot Four**, **Find Four**, **Fourplay**, **Four in a Row** and **Four in a Line**) is a two-player game in which the players first choose a colour and then take turns dropping their colour discs from the top into a seven-column, six-row vertically-suspended grid. The pieces fall straight down, occupying the next available space within the column. The object of the game is to connect four of one's own discs of the same colour next to each other vertically, horizontally, or diagonally before one's opponent can do so.

Does this sound pretty simple..!!?? So here comes the twist. No more boring gameplay with only the normal colour disks falling to their places. If u fell that you made a bad move or your opponent played a tricky move then you will get a chance to ruin his strategy by using your powers. In this game of **power connect 4** you will get 4 types of special powers, one of each of them per set.

So what do you say, now can u overpower your opponent. Can you line up four discs in a row before your opponent does?

Problem Statement:

The teams have to submit codes for an artificial intelligence which can play a complete game of power connect 4. At a time, two codes will play the game of power connect 4.

Framework:

The competition aims to provide an interface for codes of different teams to simultaneously play the modified version of the famous game 'CONNECT FOUR' which is also know by the name 'FOUR IN A ROW' against each other. The codes can be in the following languages:

1. C
2. C++
3. Java
4. Python

Note: In the entire documentation, *'user'* refers to the team and *'admin'* refers to the administrator. A part of the code will be submitted by the user and the rest will be created

by the admin to help simulate and execute the entire game. The position of the disks present in frame will be given as integers (completely explained in the format of board.txt file later in this document).

Game Explanation:

The game is a modified version of connect 4.The only difference is the extra 4 types of special disk.

1. **Clear row**
2. **Clear column**
3. **Clear first neighbours**
4. **Dual colour**

1.) **Clear row:** A self-destructing disk which will first go to the lower-most unoccupied place available in the selected column and will destroy each and every disk in that row including itself.

2.) **Clear column:** A self-destructing disk which will first go to the lower-most unoccupied place available in the selected column and will destroy each and every disk in that column including itself.

3.) **Clear first neighbours:** A self-destructing disk which will first go to the lower-most unoccupied place available in the selected column and will destroy each and every first neighbour disk in that row including itself.

e.g. if the **Clear first neighbours** disk is used with the 4^{th} column selected and suppose 11 (see fig. below) is the lowest unoccupied position then it will destroy the disks at 17,10,3,4,5,12 and 19 if they are present and after that the disk will destroy itself.

4.) **Dual colour:** This is the NEUTRAL disk. It will support both the colours.

NOTE: Both teams will have all the 4 powers total but can use only once in a set anytime during the set but the DUAL COLOUR DISK must be used in the **first ten turns** of the team else the move will be consider illegal. Except the dual colour power, using the rest of the three powers is not a compulsion.

User Task:

The user has to make a program named <Teamid>.<file_extension>. Suppose the Team Id of a particular team is Co0023, then the program name should be Co0023.cpp or Co0023.java as the case may be. This program made by user will represent his decision at a particular stage of the game. The codes will have to produce the output.txt file when they are called.

The first integer stands for the type of disk they want to use and the other integer stand for indicating the column in which the team wish to place their disc.

The numbers are in the same order as the spaces in the board.txt file i.e. 1 for the leftmost column and 7 for the rightmost column.

Text Files:

To guide the user the entire course of game being played, he will be provided with some text files. Below is the description of them:

team_no.txt: This text file will show the team number (1 or 2). The code has to read its team no. from this file for playing.

board.txt: This text file will show the positions of the discs already placed in the frame and the places which are still blank. This file will be having 42 integers separate d by blank spaces.

0 – Indicates that the place is unoccupied.

1 – Indicates that team number 1 got its disc there.

2 – Indicates that the team number 2 got its disc there.

12 – Indicates that the dual colour block is present there.

The 42 integers in the file will be for their position on board as shown in fig.

output.txt: This is the file which the user's code has to produce to tell their decision to the admin about their next move.

The team's code output will be 2 integers in this text file. **First integer** to choose the type of disk they want to use and the **second integer** to choose the column in which they want to place their disk.

The range of the **First integer** is from 1-5

1-Clear row disk

2-Clear column disk

3-Clear first neighbours disk

4-Dual colour disk

5-Normal disk

The range of **Second integer** is 1-7

1 For the leftmost column and 7 for the rightmost column

Apart from these a team can create its own help file for analysing opponent's gameplay or deciding their gameplay but not more than 5 files at a time.

Editable permissions to files:

All the files are not both readable and writable. The permissions goes as follows:

1. **team_no.txt:** only readable
2. **board.txt:** only readable.
3. **output.txt:** readable and writable. The team has to write or return its decision in this file only.

The above files will be made by the administrator and will be in the same folder as the main program of the team. So, the teams can directly use the names of these files in their codes. The help files will be made by the teams only for their own convenience itself. These files can either be in the current folder or can be inside a new sub-directory created by the team in the current folder. Teams should create their help files (if needed) with names starting with their team id. For example Co0012.txt so that it can be seen which file is created by which team code and also this is to avoid common help file names of 2 codes resulting in both editing the same file.