AUCSC 111 Homework Project
< Gomoku Game>

The homework assignment has the same late submission policy as lab.

Late submission within 24 hours: 10 points deduction
Late submission within 48 hours: 20 points deduction
After that: no points

Project-1 deadline : Monday, November 11th. (Submit gomoku1.py)
Project-2 deadline : Monday, December 9th (Submit gomoku2.py)

==========Project 1 ==========

In this homework, we will make a game **Gomoku**, also called **Five in a Row**. Gomoku is a board game that is played on a grid, so we'll use a Cartesian coordinate system with XY coordinates. It is a game played with two players (in your game, computer and you).

The description of the Gomoku below can be different from a traditional Gomoku. Please follow the instruction below for this project:
Our Gomoku has *m* x *n* board (m and n can be different) and the stone 'O' will be used for a human, and 'X' will be used for a computer.

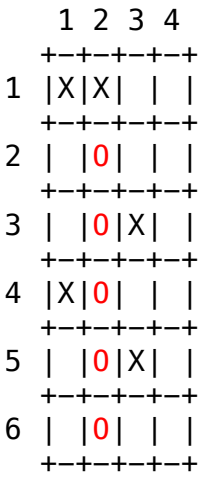
1. Rule description

You and computer put one stone alternatively. The winner is the first player to form **an unbroken chain of five stones horizontally, vertically, or diagonally**.

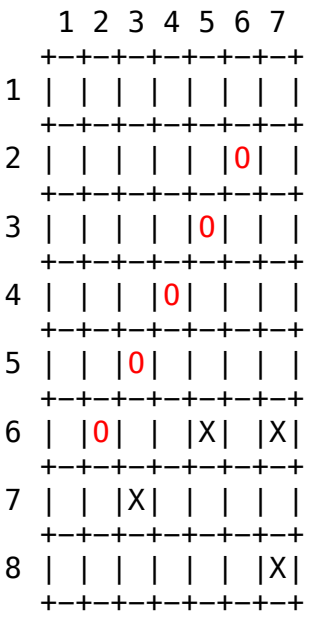
1.1 Unbroken chain of five stones horizontally – e.g.)

```
   1 2 3 4 5 6 7 8 9
  +-+-+-+-+-+-+-+-+-+
1 | |X| | |X| | | | |
  +-+-+-+-+-+-+-+-+-+
2 | | | | | | | | | |
  +-+-+-+-+-+-+-+-+-+
3 |X| | | | | | | | |
  +-+-+-+-+-+-+-+-+-+
4 | | | | | | | | | |
  +-+-+-+-+-+-+-+-+-+
5 |O|O|O|O|O| | | | |
  +-+-+-+-+-+-+-+-+-+
6 | | | | | | | | |X|
  +-+-+-+-+-+-+-+-+-+
7 | | | | | | | | | |
  +-+-+-+-+-+-+-+-+-+
8 | | | | | | | | | |
  +-+-+-+-+-+-+-+-+-+
9 | | | | | | | | | |
  +-+-+-+-+-+-+-+-+-+
```

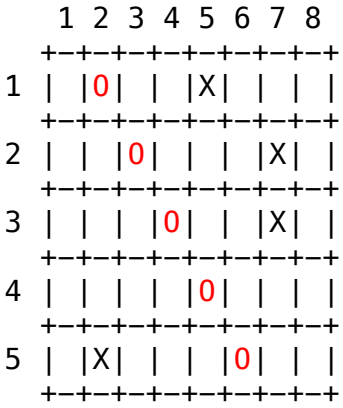
1.2. Unbroken chain of five stones vertically – e.g.)



1.3. Unbroken chain of five stones diagonally (right) – e.g.)



1.4 Unbroken chain of five stones diagonally (left) – e,g,)



2. Executing Python program

2.1.Your program name will be gomoku1.py. If you execute '**python gomoku1.py'**, then your program 'gomoku1.py' asks you about the size of the board as shown below:

```
Miyoungs-MacBook-Pro:AUCSC111 colorful$ python gomoku1.py
Input the number of rows that you want:(1-10) 10
Input the number of rows that you want:(1-10) 8
```

As an example, '10' and '8' were chosen for rows and cols.

2.2. Then, your program will display the corresponding empty board:

```
Miyoungs-MacBook-Pro:AUCSC111 colorful$ python gomoku1.py
Input the number of rows that you want:(1-10) 10
Input the number of rows that you want:(1-10) 8
   1 2 3 4 5 6 7 8
  +-+-+-+-+-+-+-+-+
1 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
2 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
3 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
4 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
5 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
6 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
7 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
8 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
9 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
10| | | | | | | | |
  +-+-+-+-+-+-+-+-+
```

2.3. You(User) begin to put a stone first. So, your program asks you about where the stone will be put. Your program will let you know exactly the boundary of the number that you can choose according to the size of the board.
e.g.)
Input The ROW where you want to put your stone **between 1 and 10:**
Input The COL where you want to put your stone **between 1 and 8:**

```
   1 2 3 4 5 6 7 8
  +-+-+-+-+-+-+-+-+
1 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
2 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
3 | | | | | | | | |
```

```
  +-+-+-+-+-+-+-+-+
4 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
5 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
6 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
7 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
8 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
9 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
10| | | | | | | | |
  +-+-+-+-+-+-+-+-+
```

```
===============
Human"s Turn
===============
Input The ROW where you want to put your stone between 1 and 10:5
Input The COL where you want to put your stone between 1 and 8:4
```

As an example, ‘5’ and ‘4’ were chosen.

2.4. Then, your program will describe the position that was chosen, then put a stone ‘O’ in the position, and show the changed board. After showing a changed board, your program will print ‘**Press 'c' to continue for machine's turn:**’. Then, when input ‘c’ is given, machine’s turn will start.

```
===============
Human"s Turn
===============
Input The ROW where you want to put your stone between 1 and 10:5
Input The COL where you want to put your stone between 1 and 8:4
```

```
===============
Human"s Choice is 5,4
===============
   1 2 3 4 5 6 7 8
  +-+-+-+-+-+-+-+-+
1 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
2 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
3 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
4 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
5 | | | |O| | | | |
  +-+-+-+-+-+-+-+-+
6 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
```

```
7 | | | | | | | | |
  +-+-+-+-+-+-+-+-+-+
8 | | | | | | | | |
  +-+-+-+-+-+-+-+-+-+
9 | | | | | | | | |
  +-+-+-+-+-+-+-+-+-+
10| | | | | | | | |
  +-+-+-+-+-+-+-+-+-+
Press 'c' to continue for machine's turn: c
```

2.5. Now, it's a machine's turn. The program shows a machine's choice which was decided randomly, and the chosen spot will be occupied with 'X'. Then, a human's turn will come.

```
===============
Random Machine"s Turn
===============
===============
Random Machine"s Choice is 1,4
===============
   1 2 3 4 5 6 7 8
  +-+-+-+-+-+-+-+-+-+
1 | | | |X| | | | | |
  +-+-+-+-+-+-+-+-+-+
2 | | | | | | | | |
  +-+-+-+-+-+-+-+-+-+
3 | | | | | | | | |
  +-+-+-+-+-+-+-+-+-+
4 | | | | | | | | |
  +-+-+-+-+-+-+-+-+-+
5 | | | |O| | | | |
  +-+-+-+-+-+-+-+-+-+
6 | | | | | | | | |
  +-+-+-+-+-+-+-+-+-+
7 | | | | | | | | |
  +-+-+-+-+-+-+-+-+-+
8 | | | | | | | | |
  +-+-+-+-+-+-+-+-+-+
9 | | | | | | | | |
  +-+-+-+-+-+-+-+-+-+
10| | | | | | | | |
  +-+-+-+-+-+-+-+-+-+

===============
Human"s Turn
===============
Input The ROW where you want to put your stone between 1 and 10:
```

2.6. Let's suppose that human chose (4,3) and the stone 'O' was put at the position.

```
===============
Human"s Turn
===============
```

```
Input The ROW where you want to put your stone between 1 and 10:4
Input The ROW where you want to put your stone between 1 and 8:3
===============
Human"s Choice is 4,3
===============
   1 2 3 4 5 6 7 8
  +-+-+-+-+-+-+-+-+
1 | | | |X| | | | |
  +-+-+-+-+-+-+-+-+
2 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
3 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
4 | | |O| | | | | |
  +-+-+-+-+-+-+-+-+
5 | | | |O| | | | |
  +-+-+-+-+-+-+-+-+
6 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
7 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
8 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
9 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
10| | | | | | | | |
  +-+-+-+-+-+-+-+-+
Press 'c' to continue for machine's turn: c
```

2.7. The machine's choice was (1,3). Then, human's turn will come again.

```
===============
Random Machine"s Turn
===============
===============
Random Machine"s Choice is 1,3
===============
   1 2 3 4 5 6 7 8
  +-+-+-+-+-+-+-+-+
1 | | |X|X| | | | |
  +-+-+-+-+-+-+-+-+
2 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
3 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
4 | | |O| | | | | |
  +-+-+-+-+-+-+-+-+
5 | | | |O| | | | |
  +-+-+-+-+-+-+-+-+
6 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
7 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
```

```
8 | | | | | | | | |
  +-+-+-+-+-+-+-+-+-+
9 | | | | | | | | |
  +-+-+-+-+-+-+-+-+-+
10| | | | | | | | |
  +-+-+-+-+-+-+-+-+-+
===============
Human"s Turn
===============
Input The ROW where you want to put your stone between 1 and 10:
```

This process will continue until all spots are taken.

2.8. In the project-1, you finish the program when all spots are taken and no more empty spots are left. In the project-1, you don't need to calculate the number of stones in a row. So, we will not determine who won in the project-1.

```
===============
Random Machine"s Choice is 4,2
===============
   1 2 3 4 5 6 7 8
  +-+-+-+-+-+-+-+-+-+
1 |X|X|X|X|O|O|X|O|
  +-+-+-+-+-+-+-+-+-+
2 |O|X|O|X|O|O|X|X|
  +-+-+-+-+-+-+-+-+-+
3 |X|X|O|O|O|O|X|O|
  +-+-+-+-+-+-+-+-+-+
4 |O|X|O|O|O|O|X|O|
  +-+-+-+-+-+-+-+-+-+
5 |X|O|X|O|O|X|O|X|
  +-+-+-+-+-+-+-+-+-+
6 |X|O|O|X|O|X|O|X|
  +-+-+-+-+-+-+-+-+-+
7 |O|O|X|O|O|X|O|X|
  +-+-+-+-+-+-+-+-+-+
8 |X|O|X|X|X|O|X|X|
  +-+-+-+-+-+-+-+-+-+
9 |O|O|X|O|X|O|X|O|
  +-+-+-+-+-+-+-+-+-+
10|X|O|X|X|O|X|X|X|
  +-+-+-+-+-+-+-+-+-+
************************
No more spot left. Game Ended.
************************
```

3. Python code description

(1) The numbers of rows, and columns for the board size, and the variable board[] will be global variables (These variables should be declared out of a function).

If you have m rows, and n columns, then 2D-board initialization should be done as following:

```
board = [[' ']*n for i in range(m)]
```

In this board initiation, all board spots will be initialized by ' '.

(2) Your "gomoku1.py" should be like this:

```
# at this position, implement code which gets the board size (the numbers
of rows and columns) from the user.
#then, initialize board here according to the board size


Print_Board() # show the board according to the size
while(True):
    Human_Turn() # get human's input to put a stone 'O'.
   if(All_Spots_Chosen()):
                print("**************************")
                print("No more spot left. Game Ended.")
                print("**************************")
                break

   #At this position, implement the code which gets 'c' key to continue
for a machine's turn.

    Random_Machine_Turn()# Machin randomly choose a spot to put a stone
'X'.
   if(All_Spots_Chosen()):
                print("**************************")
                print("No more spot left. Game Ended.")
                print("**************************")
                break
```

It means, your code should have functions **Print_Board(), Human_Turn(), All_Spots_Chosen(), and Random_Machine_Turn().**

- Print_Board() : show the current board
- Human_Turn() : get inputs (XY coordinates) from a user, and put a stone 'O' on the chosen spot. When a user's input is weird (e.g., 'return' key, number less than 1, number greater than 10, input which is not a number), then say 'wrong input' and then request for the input again. If the user's input spot is not empty, then say 'The spot has been already taken. Please choose another spot.' And then request for the input again. After that, Print_Board() function should be called to display the changed board.

e.g.)

```
===============
Random Machine"s Choice is 4,4
===============
   1 2 3 4 5
  +-+-+-+-+-+
1 |O| | | | |
```

```
  +-+-+-+-+-+
2 | | | | | |
  +-+-+-+-+-+
3 | | | | | |
  +-+-+-+-+-+
4 | | | |X| |
  +-+-+-+-+-+
5 | | | | | |
  +-+-+-+-+-+
===============
Human"s Turn
===============
Input The ROW where you want to put your stone between 1 and 5:0
Wrong Input.
Input The ROW where you want to put your stone between 1 and 5:6
Wrong Input.
Input The ROW where you want to put your stone between 1 and 5:345
Wrong Input.
Input The ROW where you want to put your stone between 1 and 5:cdf
Wrong Input.
Input The ROW where you want to put your stone between 1 and 5:
Wrong Input.
Input The ROW where you want to put your stone between 1 and 5:1
Input The COL where you want to put your stone between 1 and 5:1
The spot (1,1) has been already taken!!
Input The ROW where you want to put your stone between 1 and 5:4
Input The COL where you want to put your stone between 1 and 5:4
The spot (4,4) has been already taken!!
Input The ROW where you want to put your stone between 1 and 5:
```

- Random_Machine_Turn() : Machine chooses a spot randomly. If the machine's chosen spot is not empty, then say 'the spot ([write the chosen row number here], [write the chosen col number here]) has been already taken!! Press 'c' key if you want machine to choose a random spot again!!!'. Machine will choose another spot randomly if 'c' is entered. You have to request for the input until 'c' has been given from the user correctly.

e.g.)

```
===============
Human"s Choice is 4,2
===============
   1 2 3 4 5 6 7
  +-+-+-+-+-+-+-+
1 | | | | | | | |
  +-+-+-+-+-+-+-+
2 | | | | | | | |
  +-+-+-+-+-+-+-+
3 | | | | | | |X|
  +-+-+-+-+-+-+-+
4 |X|O|O| | | |X|
  +-+-+-+-+-+-+-+
5 |O|O|O|X| | | |
  +-+-+-+-+-+-+-+
Press 'c' to continue for machine's turn: c
===============
Random Machine"s Turn
===============
The spot (4,2) has been already taken!!
Press 'c' key if you want machine to choose a random spot again!!4
Press 'c' key if you want machine to choose a random spot again!!
Press 'c' key if you want machine to choose a random spot again!!c
```

```
The spot (4,1) has been already taken!!
Press 'c' key if you want machine to choose a random spot again!!c
The spot (5,1) has been already taken!!
Press 'c' key if you want machine to choose a random spot again!!c
The spot (5,3) has been already taken!!
Press 'c' key if you want machine to choose a random spot again!!c
===============
Random Machine"s Choice is 1,6
===============
   1 2 3 4 5 6 7
  +-+-+-+-+-+-+-+
1 | | | | | |X| |
  +-+-+-+-+-+-+-+
2 | | | | | | | |
  +-+-+-+-+-+-+-+
3 | | | | | | |X|
  +-+-+-+-+-+-+-+
4 |X|O|O| | | |X|
  +-+-+-+-+-+-+-+
5 |O|O|O|X| | | |
  +-+-+-+-+-+-+-+
```

- All_Spots_Chosen() : Check if all spots in the board are taken. **If all spots are taken and no empty spots are left, return False. Otherwise, return True.**

4. Other requirements

1) When getting the size of the board from a user, the input for the number of rows(columns) should be between 1 and 10. If other numbers or non-number characters are given, show an error message and request for the input again.

e.g.)

```
Miyoungs-MacBook-Pro:AUCSC111 colorful$ python gomoku1.py
Input the number of rows that you want:(1-10) 0
Wrong Input
Input the number of rows that you want:(1-10)
Wrong Input
Input the number of rows that you want:(1-10) 11
Wrong Input
Input the number of rows that you want:(1-10) A
Wrong Input
Input the number of rows that you want:(1-10) abc
Wrong Input
Input the number of rows that you want:(1-10)
```

2) Your code should not be similar with your classmates'. Do not look at your classmates' codes.

3) Don't forget to add proper comments on the code and substrings for functions. Otherwise, you will have deductions.

==========Project-2==========

Project-2 is to add **Intelligence** to the machine of the Project-1, and to decide who wins in the game.
So, all the requirements of the Project-1 should be implemented in Project-2. If any part of the Project-1 is missing or is not working, you will have deductions.
The code name will be gomoku2.py.

1. Rule Description

1.1 After getting input for the size of the board, and then displaying the empty board, you have to print "**Who do you want to play with? 1. Random machine, 2. Smart machine:** "

e.g)

```
Miyoungs-MacBook-Pro:AUCSC111 colorful$ python gomoku2.py
Input the number of rows that you want:(1-10) 6
Input the number of rows that you want:(1:-10) 8
   1 2 3 4 5 6 7 8
  +-+-+-+-+-+-+-+-+
1 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
2 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
3 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
4 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
5 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
6 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
Who do you want to play with? 1.Random machine 2. Smart machine:
```

2. If '1' is chosen, the program will work in the same way of the Project-1. If '2' is chosen, your machine will become smart. Your program finds the position which ensures the biggest number of human stones in a row horizontally, vertically or diagonally if the position is taken by human. Then, the position will be taken by machine to prevent human from winning.

e.g.)

```
Input the number of rows that you want:(1-10) 7
Input the number of cols that you want:(1-10) 8
   1 2 3 4 5 6 7 8
  +-+-+-+-+-+-+-+-+
1 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
2 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
3 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
4 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
5 | | | | | | | | |
```

```
  +-+-+-+-+-+-+-+-+
6 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
7 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
Who do you want to play with? 1.Random machine 2. Smart machine: 2
===============
Human"s Turn
===============
Input The ROW where you want to put your stone between 1 and 7:5
Input The COL where you want to put your stone between 1 and 8:4
===============
Human"s Choice is 5,4
===============
   1 2 3 4 5 6 7 8
  +-+-+-+-+-+-+-+-+
1 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
2 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
3 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
4 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
5 | | | |O| | | | |
  +-+-+-+-+-+-+-+-+
6 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
7 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
Press 'c' to continue for machine's turn: c
===============
Smart Machine"s Turn
===============
===============
Smart Machine"s Choice is 4,3
===============
   1 2 3 4 5 6 7 8
  +-+-+-+-+-+-+-+-+
1 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
2 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
3 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
4 | | |X| | | | | |
  +-+-+-+-+-+-+-+-+
5 | | | |O| | | | |
  +-+-+-+-+-+-+-+-+
6 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
7 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
===============
Human"s Turn
===============
Input The ROW where you want to put your stone between 1 and 7:5
Input The COL where you want to put your stone between 1 and 8:3
===============
Human"s Choice is 5,3
===============
   1 2 3 4 5 6 7 8
  +-+-+-+-+-+-+-+-+
1 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
```

```
2 | | | | | | | |
  +-+-+-+-+-+-+-+-+
3 | | | | | | | |
  +-+-+-+-+-+-+-+-+
4 | | |X| | | | |
  +-+-+-+-+-+-+-+-+
5 | | |O|O| | | |
  +-+-+-+-+-+-+-+-+
6 | | | | | | | |
  +-+-+-+-+-+-+-+-+
7 | | | | | | | |
  +-+-+-+-+-+-+-+-+
Press 'c' to continue for machine's turn: c
===============
Smart Machine"s Turn
===============
===============
Smart Machine"s Choice is 5,2
===============
   1 2 3 4 5 6 7 8
  +-+-+-+-+-+-+-+-+
1 | | | | | | | |
  +-+-+-+-+-+-+-+-+
2 | | | | | | | |
  +-+-+-+-+-+-+-+-+
3 | | | | | | | |
  +-+-+-+-+-+-+-+-+
4 | | |X| | | | |
  +-+-+-+-+-+-+-+-+
5 | |X|O|O| | | |
  +-+-+-+-+-+-+-+-+
6 | | | | | | | |
  +-+-+-+-+-+-+-+-+
7 | | | | | | | |
  +-+-+-+-+-+-+-+-+
===============
Human"s Turn
===============
Input The ROW where you want to put your stone between 1 and 7:5
Input The COL where you want to put your stone between 1 and 8:5
===============
Human"s Choice is 5,5
===============
   1 2 3 4 5 6 7 8
  +-+-+-+-+-+-+-+-+
1 | | | | | | | |
  +-+-+-+-+-+-+-+-+
2 | | | | | | | |
  +-+-+-+-+-+-+-+-+
3 | | | | | | | |
  +-+-+-+-+-+-+-+-+
4 | | |X| | | | |
  +-+-+-+-+-+-+-+-+
5 | |X|O|O|O| | |
  +-+-+-+-+-+-+-+-+
6 | | | | | | | |
  +-+-+-+-+-+-+-+-+
7 | | | | | | | |
  +-+-+-+-+-+-+-+-+
Press 'c' to continue for machine's turn: c
===============
Smart Machine"s Turn
===============
===============
Smart Machine"s Choice is 5,6
```

```
===============
   1 2 3 4 5 6 7 8
  +-+-+-+-+-+-+-+-+
1 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
2 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
3 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
4 | | |X| | | | | |
  +-+-+-+-+-+-+-+-+
5 | |X|O|O|O|X| | |
  +-+-+-+-+-+-+-+-+
6 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
7 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
===============
Human"s Turn
===============
Input The ROW where you want to put your stone between 1 and 7:4
Input The COL where you want to put your stone between 1 and 8:4
===============
Human"s Choice is 4,4
===============
   1 2 3 4 5 6 7 8
  +-+-+-+-+-+-+-+-+
1 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
2 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
3 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
4 | | |X|O| | | | |
  +-+-+-+-+-+-+-+-+
5 | |X|O|O|O|X| | |
  +-+-+-+-+-+-+-+-+
6 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
7 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
Press 'c' to continue for machine's turn: c
===============
Smart Machine"s Turn
===============
===============
Smart Machine"s Choice is 6,2
===============
   1 2 3 4 5 6 7 8
  +-+-+-+-+-+-+-+-+
1 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
2 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
3 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
4 | | |X|O| | | | |
  +-+-+-+-+-+-+-+-+
5 | |X|O|O|O|X| | |
  +-+-+-+-+-+-+-+-+
6 | |X| | | | | | |
  +-+-+-+-+-+-+-+-+
7 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
===============
```

```
Human"s Turn
===============
Input The ROW where you want to put your stone between 1 and 7:3
Input The COL where you want to put your stone between 1 and 8:4
===============
Human"s Choice is 3,4
===============
   1 2 3 4 5 6 7 8
  +-+-+-+-+-+-+-+-+
1 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
2 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
3 | | | |O| | | | |
  +-+-+-+-+-+-+-+-+
4 | | |X|O| | | | |
  +-+-+-+-+-+-+-+-+
5 | |X|O|O|O|X| | |
  +-+-+-+-+-+-+-+-+
6 | |X| | | | | | |
  +-+-+-+-+-+-+-+-+
7 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
Press 'c' to continue for machine's turn: c
===============
Smart Machine"s Turn
===============
===============
Smart Machine"s Choice is 2,4
===============
   1 2 3 4 5 6 7 8
  +-+-+-+-+-+-+-+-+
1 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
2 | | | |X| | | | |
  +-+-+-+-+-+-+-+-+
3 | | | |O| | | | |
  +-+-+-+-+-+-+-+-+
4 | | |X|O| | | | |
  +-+-+-+-+-+-+-+-+
5 | |X|O|O|O|X| | |
  +-+-+-+-+-+-+-+-+
6 | |X| | | | | | |
  +-+-+-+-+-+-+-+-+
7 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
===============
Human"s Turn
===============
Input The ROW where you want to put your stone between 1 and 7:6
Input The COL where you want to put your stone between 1 and 8:4
===============
Human"s Choice is 6,4
===============
   1 2 3 4 5 6 7 8
  +-+-+-+-+-+-+-+-+
1 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
2 | | | |X| | | | |
  +-+-+-+-+-+-+-+-+
3 | | | |O| | | | |
  +-+-+-+-+-+-+-+-+
4 | | |X|O| | | | |
  +-+-+-+-+-+-+-+-+
5 | |X|O|O|O|X| | |
```

```
  +-+-+-+-+-+-+-+-+
6 | |X| |O| | | | |
  +-+-+-+-+-+-+-+-+
7 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
Press 'c' to continue for machine's turn: c
===============
Smart Machine"s Turn
===============
===============
Smart Machine"s Choice is 7,4
===============
   1 2 3 4 5 6 7 8
  +-+-+-+-+-+-+-+-+
1 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
2 | | | |X| | | | |
  +-+-+-+-+-+-+-+-+
3 | | | |O| | | | |
  +-+-+-+-+-+-+-+-+
4 | | |X|O| | | | |
  +-+-+-+-+-+-+-+-+
5 | |X|O|O|O|X| | |
  +-+-+-+-+-+-+-+-+
6 | |X| |O| | | | |
  +-+-+-+-+-+-+-+-+
7 | | | |X| | | | |
  +-+-+-+-+-+-+-+-+
===============
Human"s Turn
===============
Input The ROW where you want to put your stone between 1 and 7:
```

3. When a current stone makes a chain of five stones in a row horizontally, vertically or diagonally, then print “CONGRATULATIONS!! Machine Win!!” or “CONGRATULATIONS!! Human Win!!” according to the winner type, then terminate the program.
e.g.)

```
   1 2 3 4 5 6 7 8 9
  +-+-+-+-+-+-+-+-+-+
1 | | | | | | | | | |
  +-+-+-+-+-+-+-+-+-+
2 | | | | | | | | | |
  +-+-+-+-+-+-+-+-+-+
3 | | | | | | | | | |
  +-+-+-+-+-+-+-+-+-+
4 | | |X|X|X|O| | | |
  +-+-+-+-+-+-+-+-+-+
5 | | |X|O|O|O| | | |
  +-+-+-+-+-+-+-+-+-+
6 | | |X|O|O| | | | |
  +-+-+-+-+-+-+-+-+-+
7 | | |X|O| | | | | |
  +-+-+-+-+-+-+-+-+-+
8 | | | | | | | | | |
  +-+-+-+-+-+-+-+-+-+
9 | | | | | | | | | |
  +-+-+-+-+-+-+-+-+-+
```

```
10| | | | | | | | | |
  +-+-+-+-+-+-+-+-+-+-+
===============
Smart Machine"s Choice is 8,3
===============
   1 2 3 4 5 6 7 8 9
  +-+-+-+-+-+-+-+-+-+-+
1 | | | | | | | | | |
  +-+-+-+-+-+-+-+-+-+-+
2 | | | | | | | | | |
  +-+-+-+-+-+-+-+-+-+-+
3 | | | | | | | | | |
  +-+-+-+-+-+-+-+-+-+-+
4 | | |X|X|X|O| | | |
  +-+-+-+-+-+-+-+-+-+-+
5 | | |X|O|O|O| | | |
  +-+-+-+-+-+-+-+-+-+-+
6 | | |X|O|O| | | | |
  +-+-+-+-+-+-+-+-+-+-+
7 | | |X|O| | | | | |
  +-+-+-+-+-+-+-+-+-+-+
8 | | |X| | | | | | |
  +-+-+-+-+-+-+-+-+-+-+
9 | | | | | | | | | |
  +-+-+-+-+-+-+-+-+-+-+
10| | | | | | | | | |
  +-+-+-+-+-+-+-+-+-+-+
**********************
CONGRATULATIONS!!!! Machine Win!!!
**********************
```

e.g.2)

```
===============
   1 2 3 4 5 6 7
  +-+-+-+-+-+-+-+-+
1 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
2 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
3 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
4 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
5 | |X| |O| | | | |
  +-+-+-+-+-+-+-+-+
6 | | | |O|X| | | |
  +-+-+-+-+-+-+-+-+
7 | | | |O| | |X| |
  +-+-+-+-+-+-+-+-+
8 | | | |O| |X| | |
  +-+-+-+-+-+-+-+-+
9 | | | | | | | | |
  +-+-+-+-+-+-+-+-+
```

```
===============
Human"s Turn
===============
Input The ROW where you want to put your stone between 1 and 9:9
Input The COL where you want to put your stone between 1 and 7:4
===============
Human"s Choice is 9,4
===============
   1 2 3 4 5 6 7
  +-+-+-+-+-+-+-+
1 | | | | | | | |
  +-+-+-+-+-+-+-+
2 | | | | | | | |
  +-+-+-+-+-+-+-+
3 | | | | | | | |
  +-+-+-+-+-+-+-+
4 | | | | | | | |
  +-+-+-+-+-+-+-+
5 | |X| |O| | | |
  +-+-+-+-+-+-+-+
6 | | | |O|X| | |
  +-+-+-+-+-+-+-+
7 | | | |O| | |X|
  +-+-+-+-+-+-+-+
8 | | | |O| |X| |
  +-+-+-+-+-+-+-+
9 | | | |O| | | |
  +-+-+-+-+-+-+-+
************************
CONGRATULATIONS!!! Human Win!!!
************************
```

2. Code Description

2.1. Your “gomoku2.py” should be like this:

```
def Check_Connected_num(input_row, input_col,stone):
        ‘ This function is already complete here ‘
        Connected_num = Row_check(input_row, input_col,stone)
        if(Connected_num>=5):
                return True

        Connected_num = Column_check(input_row, input_col,stone)
        if(Connected_num>=5):
                return True

        Connected_num=Right_Diagonal_check(input_row,
input_col,stone)
        if(Connected_num>=5):
                return True
```

```
        Connected_num=Left_Diagonal_check(input_row,
input_col,stone)
        if(Connected_num>=5):
                return True
        return False


def Smart_Machine_Turn():
        print('===============')
        print('Smart Machine"s Turn')
        print('===============')

        # at this position, implement code which can choose the
spot that is best for human. Then, that spot will be chosen by
this smart machine to prevent human from winning. Then, print
board after reflecting the change. Check if putting the stone on
the spot makes the machine win.

        winner =
Check_Connected_num(max_position_row,max_position_col,'X')
        return winner



def Machine_Turn(option):
        'This function is complete here'
        if(option=='1'):
                return(Random_Machine_Turn())
        elif option=='2':
                return(Smart_Machine_Turn())
        else:
                print("ERROR")



# at this position, implement code which gets the board size (the numbers
of rows and columns) from the user.
#then, initialize board here according to the board size



Print_Board() # show the board according to the size
# At this position, get the option value between 1 and 2. 1 means a
random machine, and 2 means a smart machine.

while(True):
    winner = Human_Turn() # get human's input to put a stone 'O'. Then,
count the human stones in a row. If the number of human stones in a row
is greater than 4, then return True. Otherwise, return False.
   if(winner==True):
         print("**************************")
         print("CONGRATULATIONS!!! Human Win!!!")
         print("**************************")
```

```
            break
    if(All_Spots_Chosen()):
                print("**************************")
                print("No more spot left. Game Ended.")
                print("**************************")
                break

    #At this position, implement the code which gets 'c' key to continue
for a machine's turn.

     Winner = Machine_Turn(option)# In Machine_Turn(option), a random
machine will be a player if option is 1, and a smart machine will be a
player if option is 2.
    if(winner==True):
                print("**************************")
                print("CONGRATULATIONS!!!! Machine Win!!!")
                print("**************************")
                break
    if(All_Spots_Chosen()):
                print("**************************")
                print("No more spot left. Game Ended.")
                print("**************************")
                break
```

As you can see in the code, you have to additional implement

```
    (1)    Check_Connected_num(input_row, input_col,stone)

       """ it checks the number of stones 'stone' in a row including
       the position (input_row, input_col)
       Input: XY coordinates input_row, input_col, and stone ('X' or
       'O')
       Return: True if there are more than 4 stones 'stone' in a row
       including the position (input_row, input_col). Otherwise
       False
    (2)   Row_check(input_row, input_col,stone)
       """ it checks the number of stones 'stone' in a row
       horizontally including the position (input_row, input_col)
       Input: XY coordinates input_row, input_col, and stone ('X' or
       'O')
       Return: True if there are more than 4 stones 'stone' in a row
       horizontally including the position (input_row, input_col).
       Otherwise False

    (3)   Column_check(input_row, input_col,stone)
       """ it checks the number of stones 'stone' in a row
       vertically including the position (input_row, input_col)
       Input: XY coordinates input_row, input_col, and stone ('X' or
       'O')
       Return: True if there are more than 4 stones 'stone' in a row
       vertically including the position (input_row, input_col).
       Otherwise False
    (4)   Right_Diagonal_check(input_row, input_col,stone)
       """ it checks the number of stones 'stone' in a row
       diagonally (right) including the position (input_row,
```

```
    input_col)
    Input: XY coordinates input_row, input_col, and stone ('X' or
    'O')
    Return: True if there are more than 4 stones 'stone' in a row
    diagonally (right) including the position (input_row,
    input_col). Otherwise False
(5)   Left_Diagonal_check(input_row, input_col,stone)
    """ it checks the number of stones 'stone' in a row
    diagonally (left) including the position (input_row,
    input_col)
    Input: XY coordinates input_row, input_col, and stone ('X' or
    'O')
    Return: True if there are more than 4 stones 'stone' in a row
    diagonally (left) including the position (input_row,
    input_col). Otherwise False


(6) Smart_Machine_Turn() # See above
(7) Random_Machine_Turn()
"""In addition to your constructed    Random_Machine_Turn() in
Project-1, please add
     winner = Check_Connected_num(machine_row, machine_col,'X')
      return winner"""
```

3. Other Requirements.

3.1 If another option value is given which is not '1' or '2', then you have to print '"Wrong input. Please choose between 1 and 2 as an option." Then, please get an input again until the correct input is given.

e.g.)

```
Who do you want to play with? 1.Random machine 2. Smart machine: 3
Wrong input.
Please choose between 1 and 2 as an option.
Who do you want to play with? 1.Random machine 2. Smart machine:
Wrong input.
Please choose between 1 and 2 as an option.
Who do you want to play with? 1.Random machine 2. Smart machine: cs
Wrong input.
Please choose between 1 and 2 as an option.
Who do you want to play with? 1.Random machine 2. Smart machine:
```

3.2. Don't forget to add substrings for functions and proper comments.

3.3. You should add at least one function on your own.