1. Develop the classic game tic-tac-toe in Python (100pts)

(a) Introduction

It is a game with two players who take turns to place chesses 'X' and 'O' in a board of 3×3 grid. The player who is the first placing three 'X' or 'O' in a horizontal, vertical, or diagonal row wins the game. The player, who goes first, always uses 'X'.

If you are new to the game, go to https://playtictactoe.org/

(b) The Task

Some functions have been defined with clear descriptions of parameters and returns, which will help you implement the game. Please read the file "Assignment2_tic_ tac_toe.py" for the details. Your task is to implement those functions in that file.

DrawBoard	5pts
IsSpaceFree	5pts
GetNumberOfChessPieces	5pts
IsBoardFull	5pts
IsBoardEmpy	5pts
UpdateBoard	5pts
HumanPlayer	10pts
ComputerPlayer	10pts if using random choice
	20pts if thinking one-step ahead
Judge	10pts
ShowOutcome	10pts
TicTacToeGame	20pts

There are two players: one player is a human user represented by the function "HumanPlayer", and the other player is the function "ComputerPlayer".

"HumanPlayer" will parse user input from the keyboard, and it should be able to handle random input string from the human user.

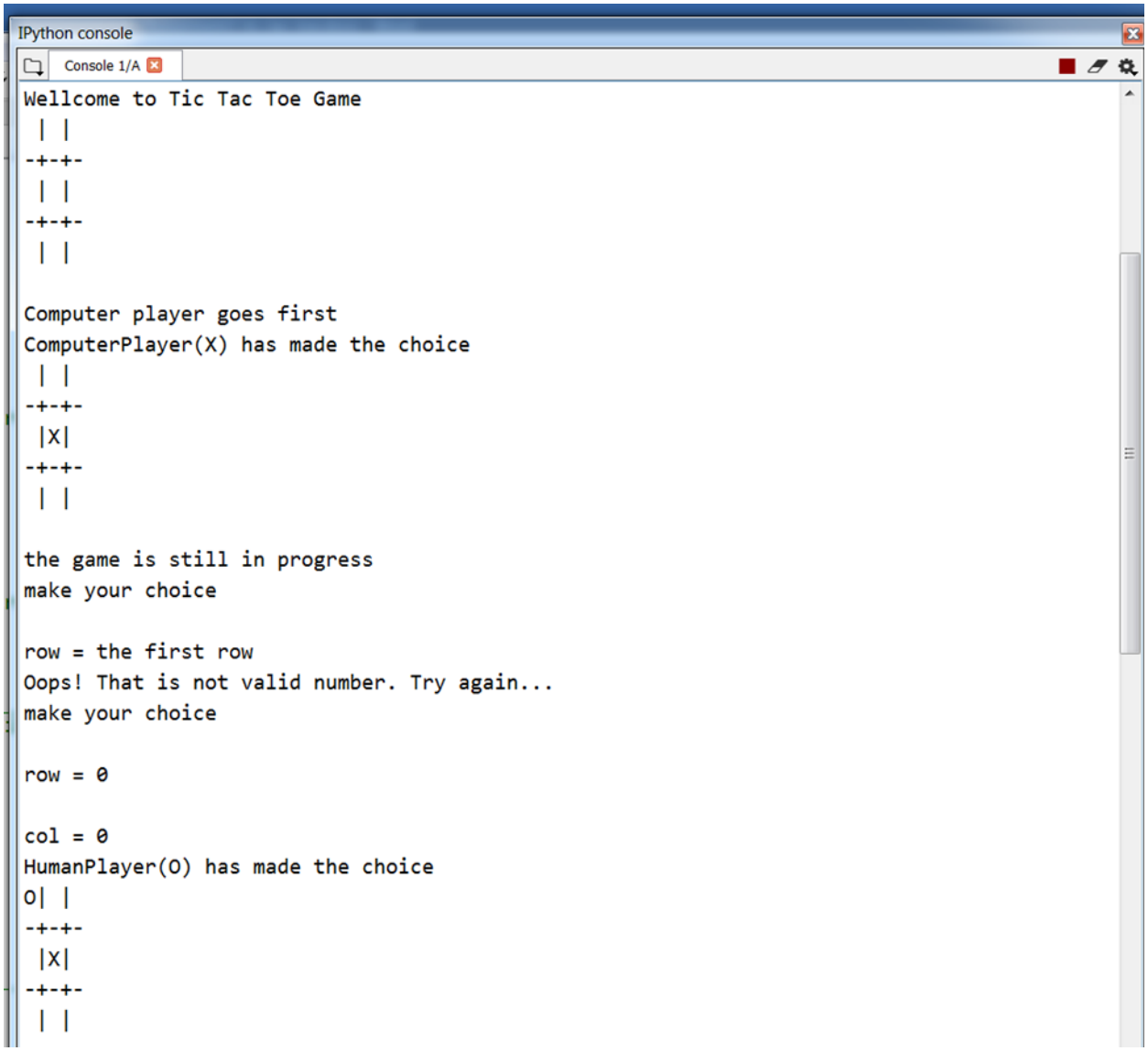
"ComputerPlayer" will use a random strategy: select a spot randomly to place a chess.

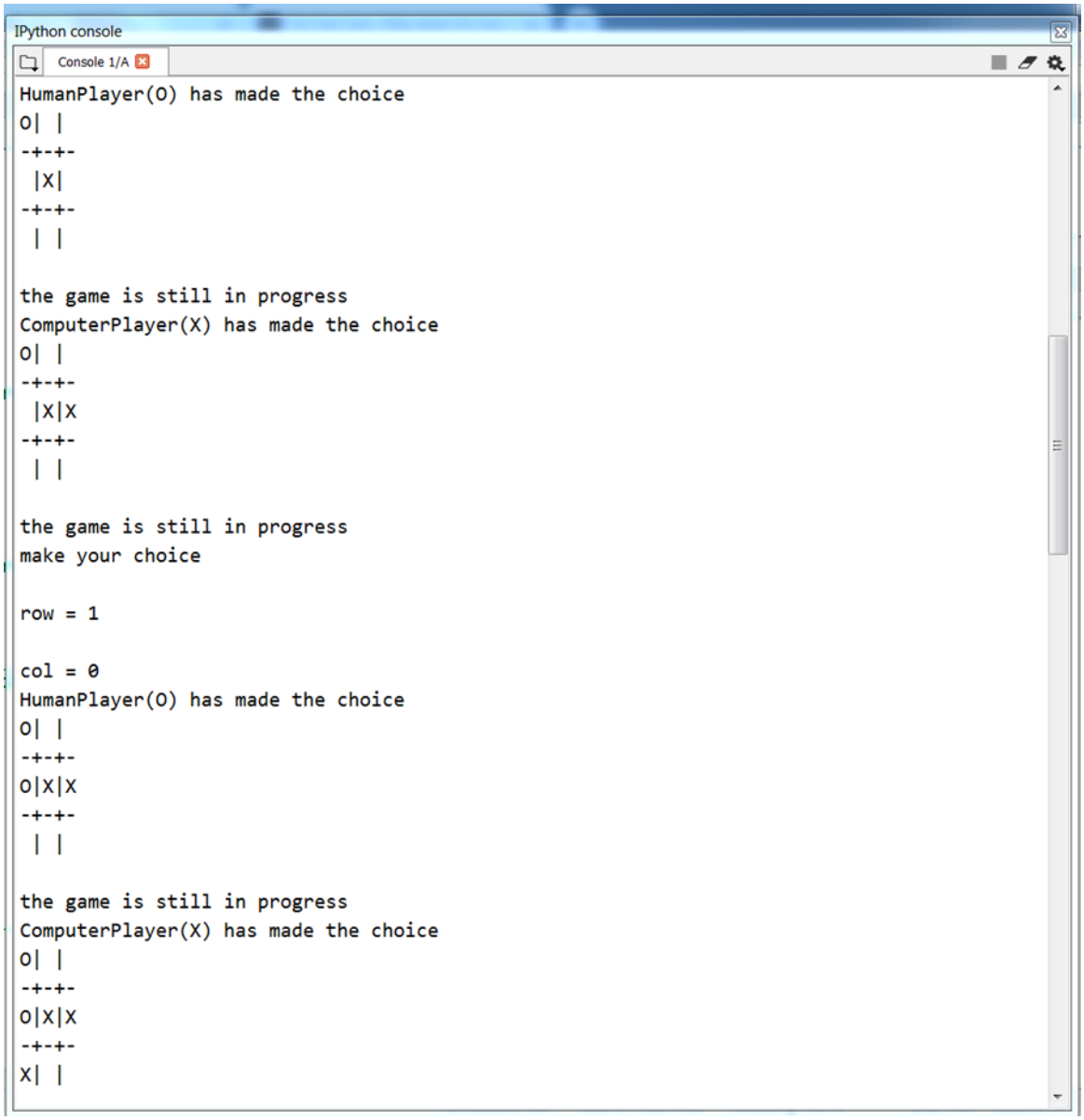
(c) Notes

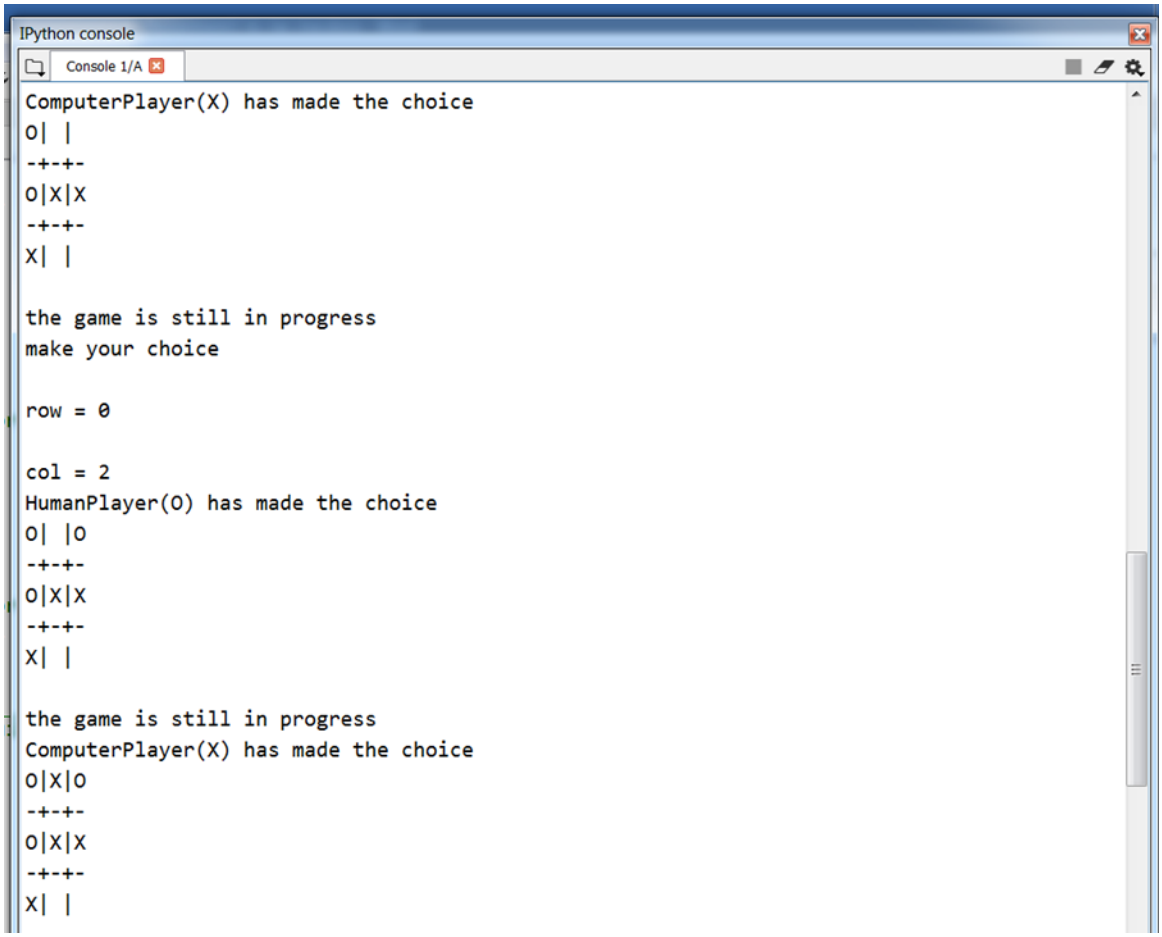
Do not use any global variables (variables outside the functions) because they are evil.
Do not change the definitions of the functions in "tic_ tac_toe.py"
You may define new functions to be used in ComputerPlayer.

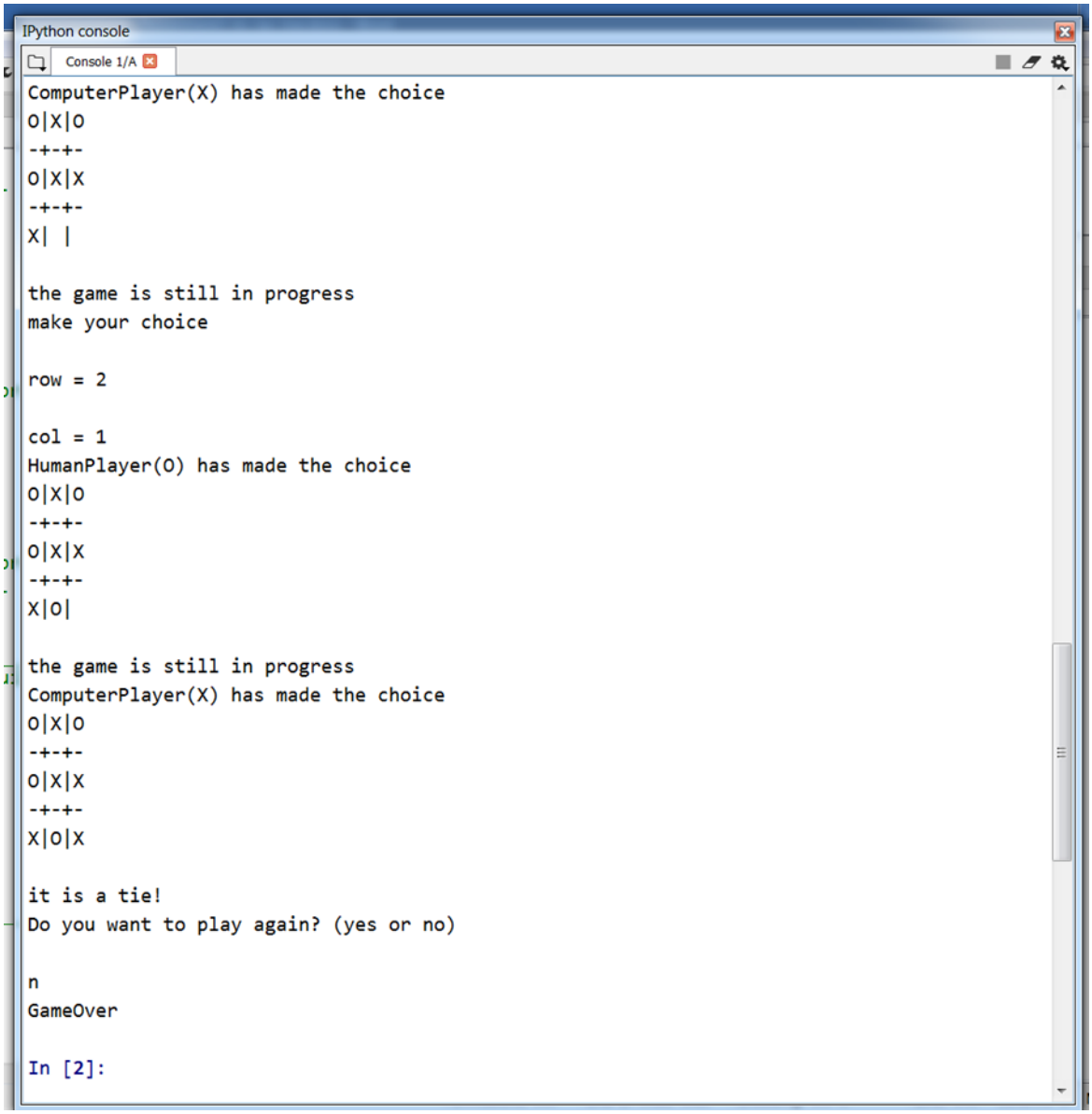
Read the sample run in the next page carefully. You are free to modify messages as long as they provide the same information to the user.

Play your game many times to make sure it has no bugs.


IPython console
Console 1/A
Wellcome to Tic Tac Toe Game
| |
-+-+-
| |
-+-+-
| |
Computer player goes first
ComputerPlayer(X) has made the choice
| |
-+-+-
|X|
-+-+-
| |
the game is still in progress
make your choice
row = the first row
Oops! That is not valid number. Try again...
make your choice
row = 0
col = 0
HumanPlayer(O) has made the choice
O| |
-+-+-
|X|
-+-+-
| |


IPython console
Console 1/A
HumanPlayer(O) has made the choice
O| |
-+-+-
|X|
-+-+-
| |
the game is still in progress
ComputerPlayer(X) has made the choice
O| |
-+-+-
|X|X
-+-+-
| |
the game is still in progress
make your choice
row = 1
col = 0
HumanPlayer(O) has made the choice
O| |
-+-+-
O|X|X
-+-+-
| |
the game is still in progress
ComputerPlayer(X) has made the choice
O| |
-+-+-
O|X|X
-+-+-
X| |


IPython console
Console 1/A
ComputerPlayer(X) has made the choice
O| |
-+-+-
O|X|X
-+-+-
X| |
the game is still in progress
make your choice
row = 0
col = 2
HumanPlayer(O) has made the choice
O| |O
-+-+-
O|X|X
-+-+-
X| |
the game is still in progress
ComputerPlayer(X) has made the choice
O|X|O
-+-+-
O|X|X
-+-+-
X| |


IPython console
Console 1/A
ComputerPlayer(X) has made the choice
O|X|O
-+-+-
O|X|X
-+-+-
X| |
the game is still in progress
make your choice
row = 2
col = 1
HumanPlayer(O) has made the choice
O|X|O
-+-+-
O|X|X
-+-+-
X|O|
the game is still in progress
ComputerPlayer(X) has made the choice
O|X|O
-+-+-
O|X|X
-+-+-
X|O|X
it is a tie!
Do you want to play again? (yes or no)
n
GameOver
In [2]:

Hints for "ComputerPlayer thinks one step ahead"

(1) if the board is empty, then place the chess in the corner

(2) try every empty spot on the board, select the one that will make a row of 3 ‘X’ or ‘O’

(2) try every empty spot on the board, select the one that will prevent the other player from making a row of 3 ‘X’ or ‘O’

You may need the function deepcopy from the module copy

```
import copy
Board_new=copy.deepcopy(Board)
```

2. Further enhancement of the computer player (Extra Credit 20pts)

The goal is to enhance “ComputerPlayer” so that it could think N steps ahead. N is an arbitrary number, and N >= 2.

You are free to add more functions.

Save your code to a different file: tic_tac_toe_enhanced.py