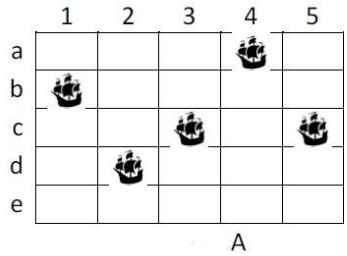
ASSIGNMENT

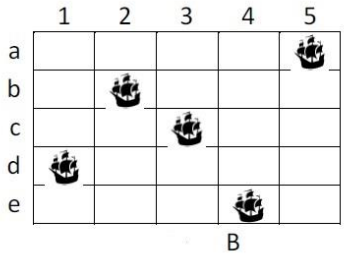
This assignment's target is to create a program in Python coding language, which plays a game of battleship. The game is played between the player and the computer or between two individual players.

INTRODUCTION - GAME DESCRIPTION

Assume a chart with 5 horizontal and 5 vertical rows, which means that the chart has 25 rectangular boxes. Each box corresponds to a marine area. In each one of these positions there can be maximum one ship (that means 0 or 1). Each player has its own chart in which there are the player's 5 ships. Each ship can take only one place at the chart, there can't be multiple ships in the same box/area. In the following image we can see an imaginarposition of each player's fleet. Player A's chart is on the left and B's on the right.



A



B

In the beginning of the game each player assigns his 5 ships in the box/area of his choice, without letting the other player know the positions. Afterwards, the two players are playing in order, and firing one shot, one at the other in each round. Each shot is falling into a box/area (for example; c3). The first player who sinks all the ships of the opponent wins.

ASSIGNMENT REQUESTS

You are assigned to write a program/code that plays a battleship game with absolute obedience to the description above. In the beginning the program will ask from the user to input 1 or 2, 1 if he wants to play with the computer, 2 if he wants to play between another individual player. For example:

```
BATTLESHIP GAME
The objective is to sink the opponent's ships before the opponent sinks yours.
Input 1 for 1-player game or 2 for 2-player game: 1
```

In each case, firstly the player 1 is called to input the positions of his 5 ships. If the game is between two players the second player must not see. If the player inputs a position that is not valid (ex. g8) or that it has been given before, the program must ask for it again. For example:

```
Player 1 enter the position of your ship no 1: a3
Player 1 enter the position of your ship no 2: c4
Player 1 enter the position of your ship no 3: a3
Invalid position, or position already taken. Try again: d2
Player 1 enter the position of your ship no 4: e1
Player 1 enter the position of your ship no 5: a2
```

If the second player is not the computer, then he is called to enter the position of his ships, without the first one watching. If the second player is the computer, the positions are picked randomly. So you can use the random and the randint (a,b) function, which generates a random integer number between a and b.

Afterwards, the game starts. The first player is randomly picked. The players are playing turn and turnabout. If the player is the human, he is asked to input the positions of enemy's ships. The computer inputs random positions. Positions that a player has inputted before, cannot be used again. It keeps which positions each player has used before, so they can't be used again. After each shot, the program prints if the shot was succesful or not and prints the map/chart of each player and the shots that they have taken from the enemy. Succesful shots are marked with 'o' and unsuccessful with 'x'. An example follows between one player (P1) and the computer (P2). The shots of player are asked by the program but the shots of the computer are picked randomly, before the beginning of the battleship, based in the description of the assignment)

```
Player 1 starts first
   P1       P2
  12345    12345
a        a
b        b
c        c
d        d
e        e
Player 1 enter the position to throw your missile: a2
Missile thrown at a2
Target missed!
   P1       P2
  12345    12345
a        a  x
b        b
c        c
d        d
e        e
Missile thrown at c5
Target missed!
   P1       P2
  12345    12345
a        a  x
b        b
c     x  c
d        d
e        e
Player 1 enter the position to throw your missile: a2
Invalid position, or missile already thrown there. Try again: f5
Invalid position, or missile already thrown there. Try again: c4
Missile thrown at c4
Target hit!
   P1       P2
  12345    12345
a        a  x
b        b
c     x  c    o
d        d
e        e
Missile thrown at e3
Target missed!
   P1       P2
  12345    12345
a        a  x
b        b
c     x  c    o
d        d
e   x    e
Player 1 enter the position to throw your missile: b2
Missile thrown at b2
Target missed!
```

```
   P1       P2
  12345    12345
a        a  x
b        b  x
c     x  c    o
d        d
e   x    e
Missile thrown at d1
Target hit!
   P1       P2
  12345    12345
a        a  x
b        b  x
c     x  c    o
d o      d
e   x    e
...
...
Player 1 enter the position to throw your missile: e3
Missile thrown at e3
Target missed!
   P1       P2
  12345    12345
a xo  x  a xxxxx
b xxx x  b xxxo
c x x x  c oxxo
d oxoxx  d xxxo
e  xxox  e xxx
Missile thrown at a4
Target hit!
   P1       P2
  12345    12345
a xo ox  a xxxxx
b xxx x  b xxxo
c x x x  c oxxo
d oxoxx  d xxxo
e  xxox  e xxx
GAME OVER. CPU wins
```

The game (and the program) ends at the time one of the players shoots successfully the last ship of the enemy (see above). Then the chart and the message are printed.

EXTRA INFORMATON

1. Save your program in the file **assignment.py**
2. Your assignment must follow directly the requests and the description of the game above. Programs that don't have absolute obedience on the requests are disqualified and they will not be marked.
3. Programs that don't run will not be marked
4. Assignments that have been copied entirely or partly from books, the World Wide Web, past and current assignments or elsewhere, will not be marked. Also, special plagiarism software and experience will be used.
5. Assignments that don't include basic commentary to the code with the hash (#) character will not be marked. You have to show to the examiner that you know what you are doing, and you are in a position to explain it.
6. Assignments that are delivered after the deadline will not be marked

Good Luck