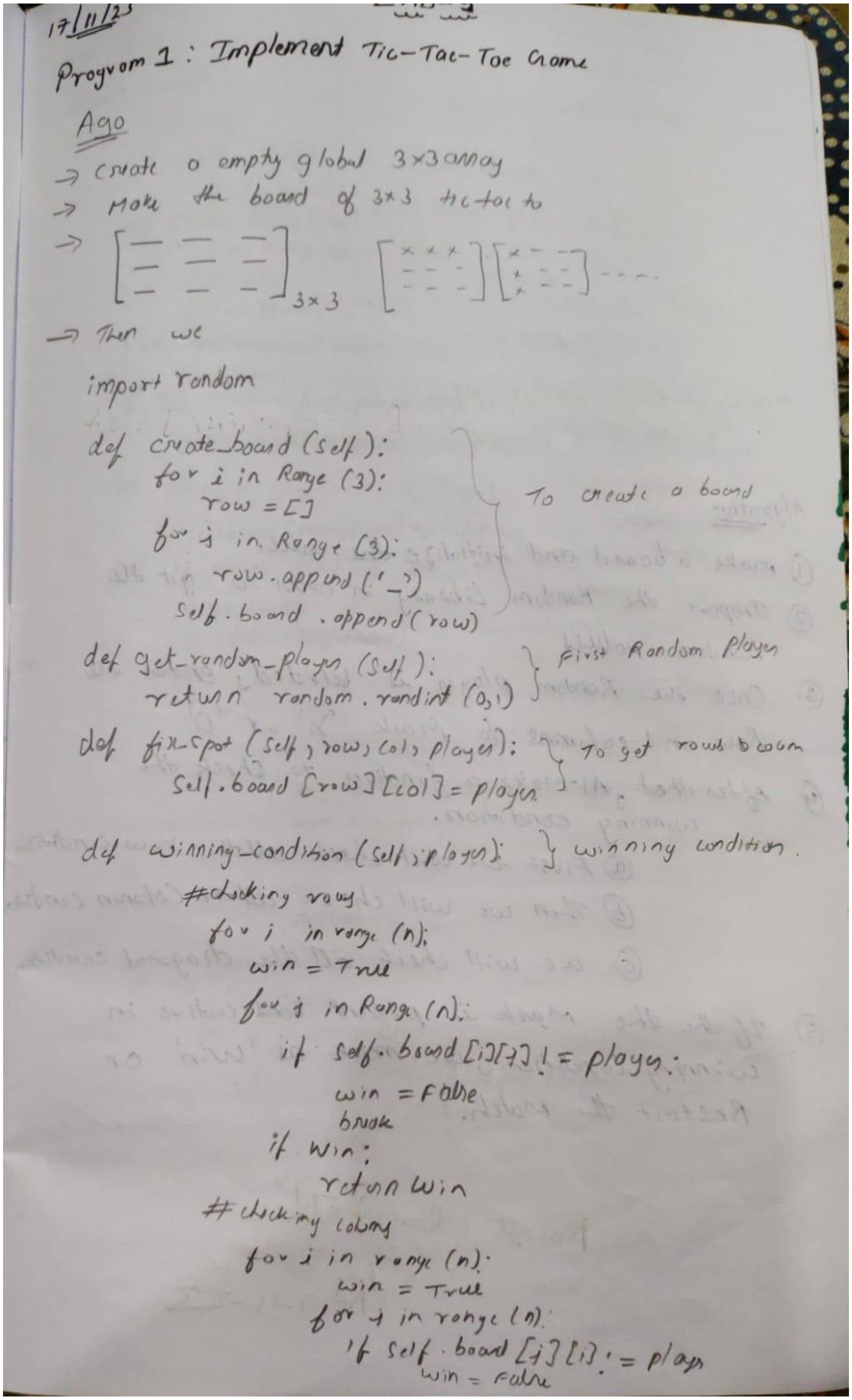


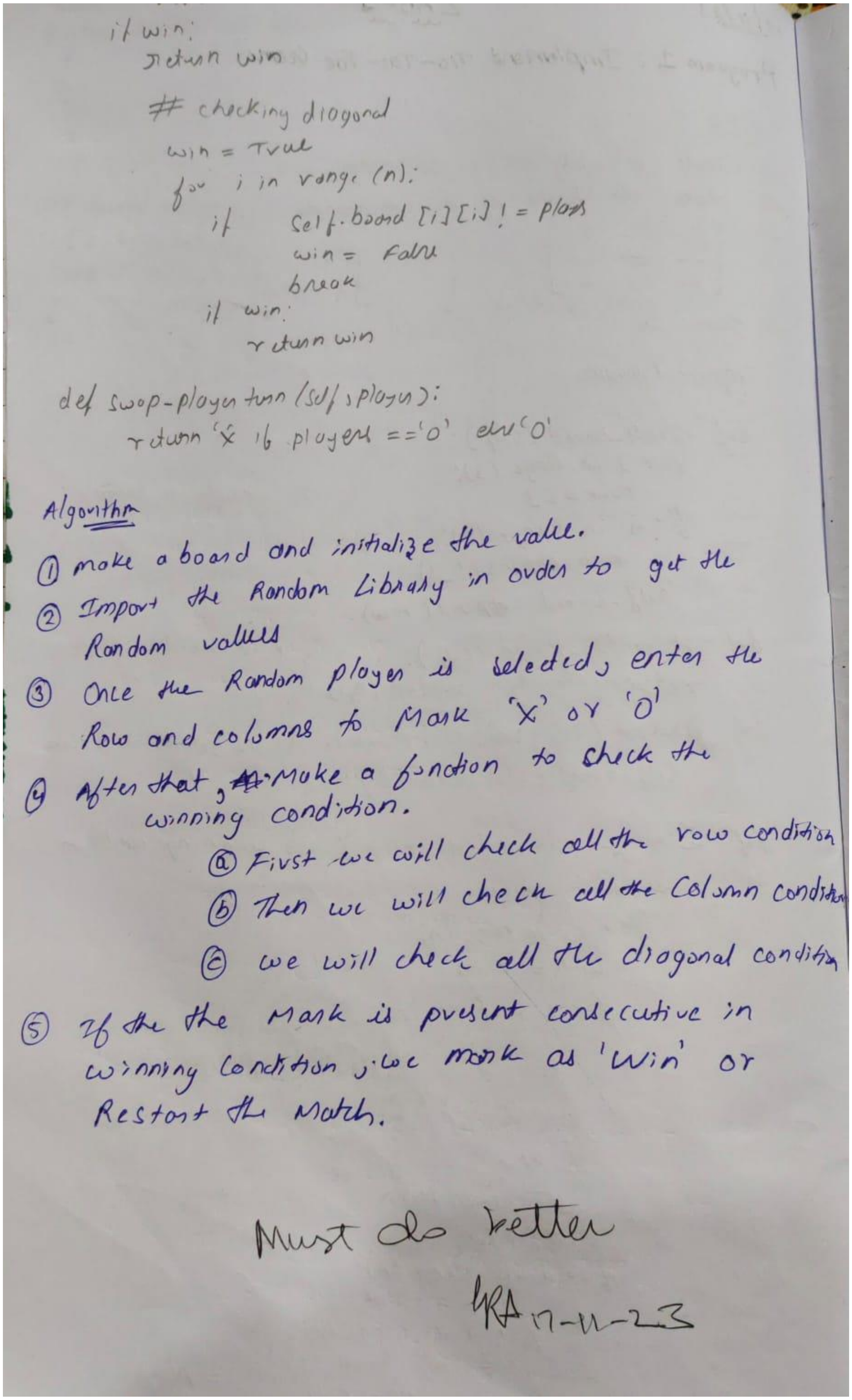
17/11/23

Program 1 : Implement Tic-Tac-Toe Game

Ago

→ create a empty global 3×3 array
→ Make the board of 3×3 tic-tac to
→ $\begin{bmatrix} - & - & - \\ - & - & - \\ - & - & - \end{bmatrix}_{3\times 3}$ $\begin{bmatrix} x & x & x \\ - & - & - \\ - & - & - \end{bmatrix}$ $\begin{bmatrix} x & - & - \\ x & - & - \\ x & - & - \end{bmatrix}$ ....
→ Then we

```
import random

def create_board (self):
        for i in Range (3):
            row = []
        for j in Range (3):                          } To create a board
            row.append ('_')
        self.board .append (row)
def get_random_player (self):                        } First Random Player
    return random.randint (0,1)
def fix_spot (self, row, col, player):               } To get rows & colum
    Sell.board [row][col] = player

def winning_condition (self, player):                } winning condition.
        #checking rows
        for i  in range (n):
            win = True
            for j in Range (n):
                if self.board [i][j] != player:
                    win = False
                    break
            if win:
                return win
        #checking colums
        for i in range (n):
            win = True
            for j in range (n):
                if self.board [j][i] != player
                win = False
```



```
if win:
  return win

   # checking diagonal
   win = True
   for i in range (n):
       if    self.board [i][i] != player
             win = False
             break
      if win:
          return win

def swap-player turn (self, player):
    return 'x' if player =='O' else 'O'
```

Algorithm

① make a board and initialize the value.

② Import the Random Library in order to get the Random values

③ Once the Random player is selected, enter the Row and columns to Mark 'X' or 'O'

④ After that, ~~A~~ Make a function to check the winning condition.

ⓐ First we will check all the row condition

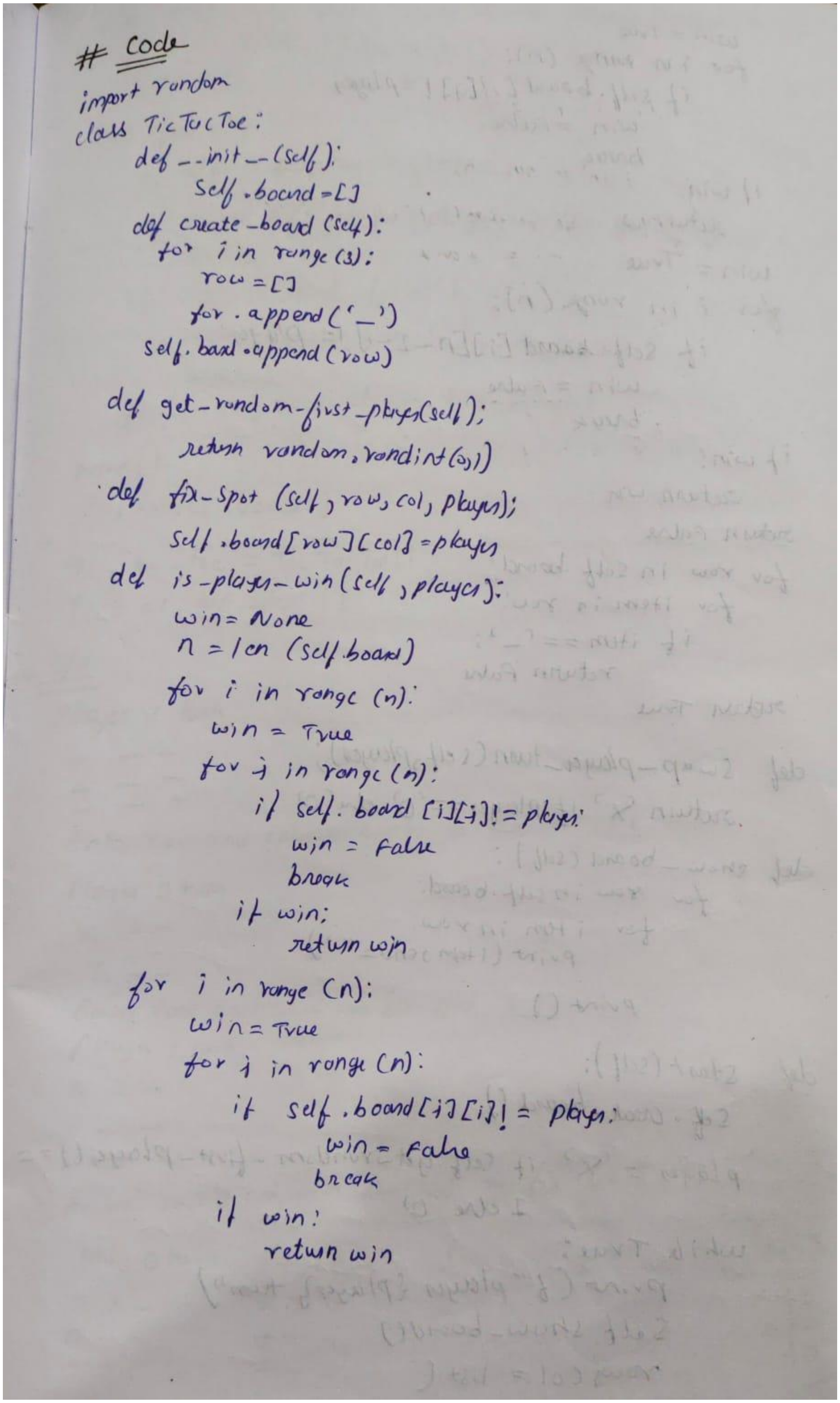
ⓑ Then we will check all the Column condition

ⓒ we will check all the diagonal condition

⑤ If the the Mark is present consecutive in winning condition, we mark as 'Win' or Restart the Match.

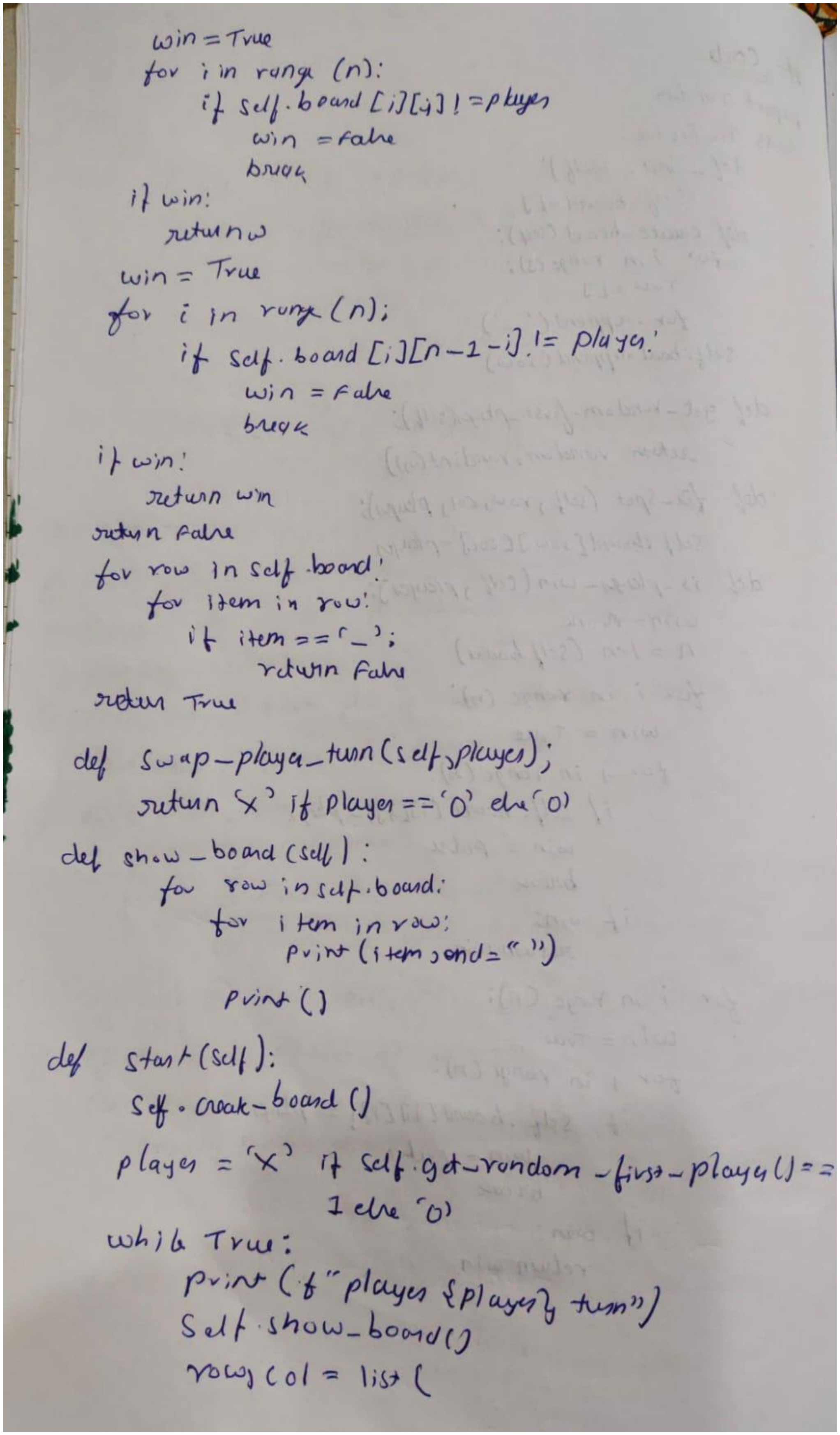
Must do better

17-11-23

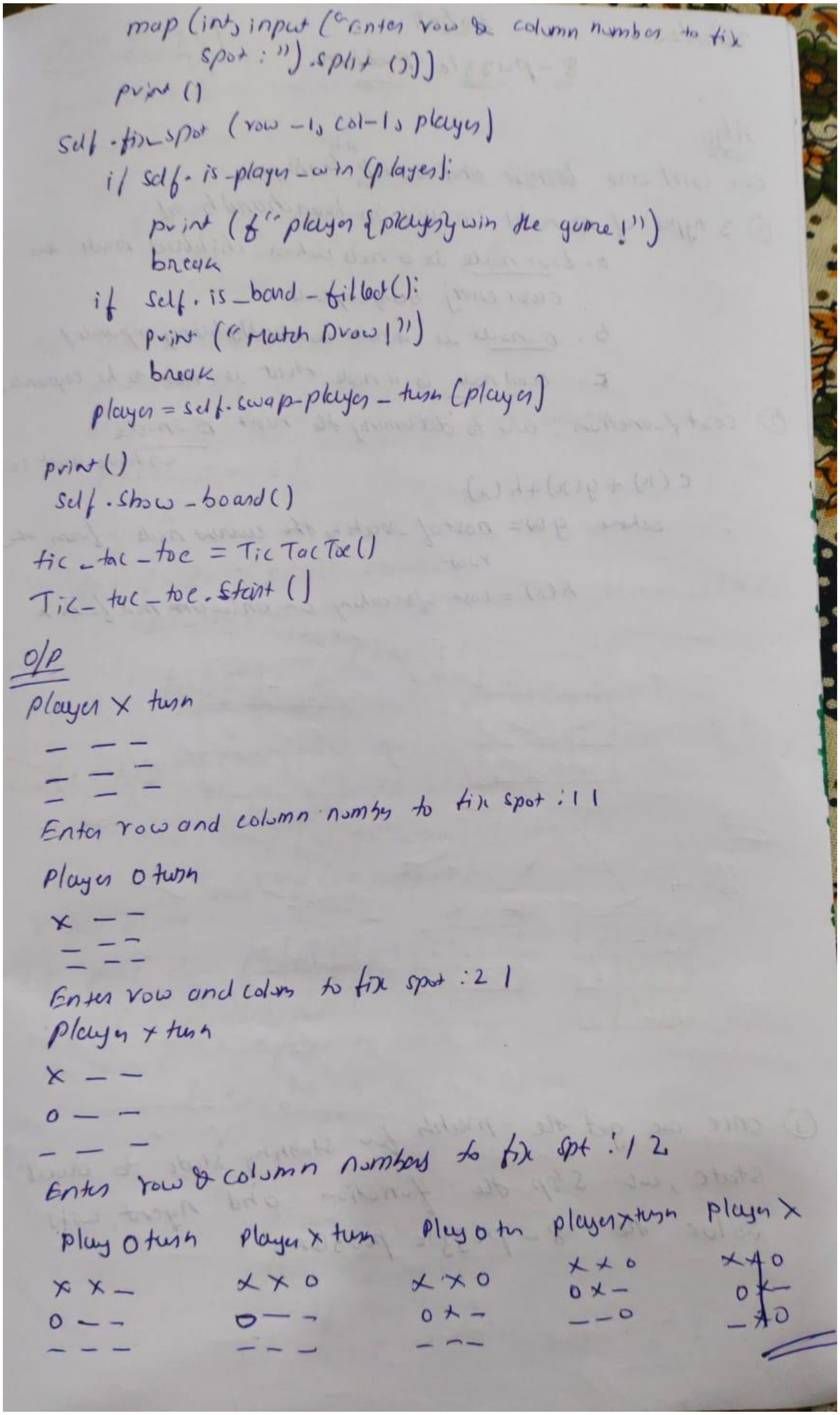


Code

```
import random
class TicTacToe:
    def __init__(self):
        Self.board=[]
    def create_board(self):
      for i in range(3):
         row=[]
         for.append('_')
      Self.bard.append(row)

    def get_random_first_player(self);
        return random,randint(0,1)
    def fix_spot (self, row, col, player);
        Self.board[row][col]=player
    def is_player_win(self, player):
        win= None
        n = len (self.board)
        for i in range (n):
           win = True
           for j in range (n):
              if self.board [i][j]!=player
                 win = False
                 break
           if win;
              return win
    for i in range (n):
       win=True
       for j in range (n):
          if self.board[j][i]!= player:
              win= False
              break
       if win:
          return win
```



```
        win=True
        for i in range (n):
            if self.board [i][i] ! =player
                win =False
                break
        if win:
            returnw
        win= True
        for i in range(n);
            if self.board [i][n-1-i].!= player.'
                win =False
                break
        if win!
            return win
        return False
        for row in self.board:
            for item in row:
                if item =='_';
                    return False
        return True

    def swap_player_turn(self,player);
        return 'x' if player =='O' else 'O'
    def show_board(self):
        for row in self.board:
            for item in row:
                print (item, end=" ")
            print ()
    def start(self):
        self.create_board()
        player = 'X' if self.get_random_first_player()==
                1 else 'O'
        while True:
            print(f"player {player} turn")
            Self.show_board()
            row, col = list(
```

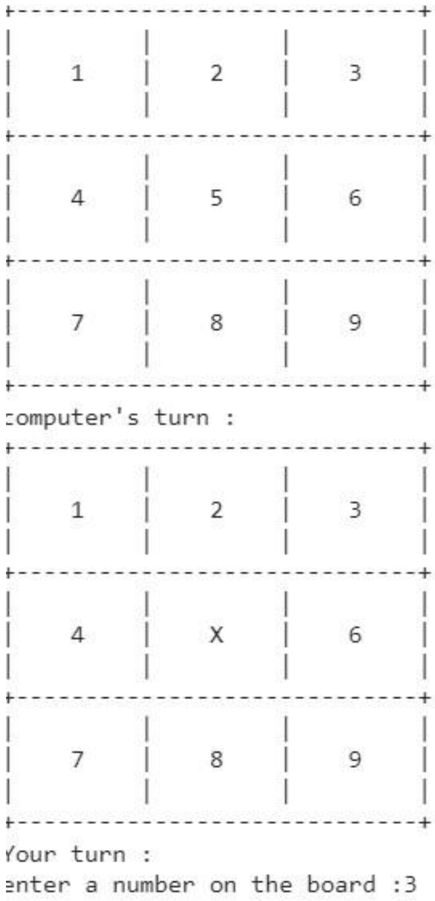


```
        map (int, input ("Enter row & column number to fix
                spot : ")).split ()))
        print ()
    self.fix_spot (row -1, col-1, player)
        if self.is_player_win (player):
            print (f"player {player} win the game!")
            break
        if self.is_band_filled():
            print ("Match Draw!")
            break
        player = self.swap_player_turn (player)

    print()
    self.show_board()

tic_tac_toe = TicTacToe()
Tic_tac_toe.start ()
```

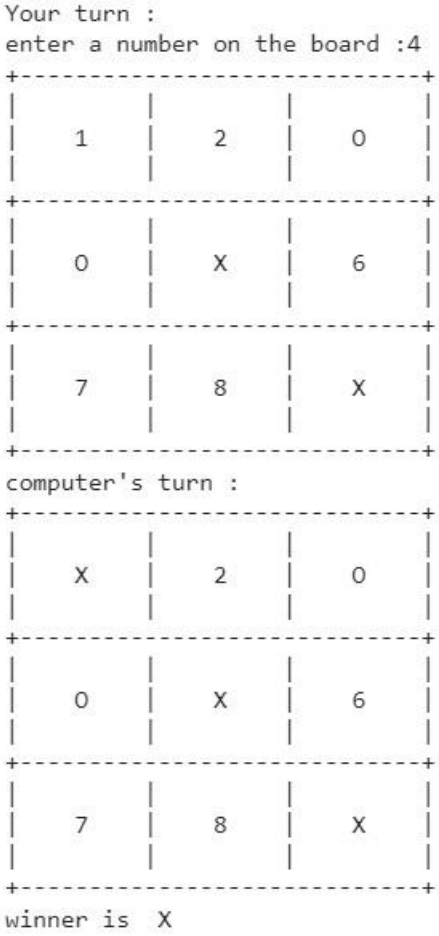
o/p

```
player X turn
 - - -
 - - -
 - - -
Enter row and column number to fix spot : 1 1
Player O turn
 X - -
 - - -
 - - -
Enter row and colum to fix spot : 2 1
player X turn
 X - -
 O - -
 - - -
Enter row & column numbers to fix spt : 1 2
 play O turn   player X turn   Play O turn   player X turn   player X
                                              X X O           X X O
 X X -          X X O           X X O         O X -           O X -
 O - -          O - -           O X -         - - O           - X O
 - - -          - - -           - - -
```

Output:



```
+-----------------------------+
|         |         |         |
|    1    |    2    |    3    |
|         |         |         |
+-----------------------------+
|         |         |         |
|    4    |    5    |    6    |
|         |         |         |
+-----------------------------+
|         |         |         |
|    7    |    8    |    9    |
|         |         |         |
+-----------------------------+
computer's turn :
+-----------------------------+
|         |         |         |
|    1    |    2    |    3    |
|         |         |         |
+-----------------------------+
|         |         |         |
|    4    |    X    |    6    |
|         |         |         |
+-----------------------------+
|         |         |         |
|    7    |    8    |    9    |
|         |         |         |
+-----------------------------+
Your turn :
enter a number on the board :3
```



```
Your turn :
enter a number on the board :4
+-----------------------------+
|         |         |         |
|    1    |    2    |    O    |
|         |         |         |
+-----------------------------+
|         |         |         |
|    O    |    X    |    6    |
|         |         |         |
+-----------------------------+
|         |         |         |
|    7    |    8    |    X    |
|         |         |         |
+-----------------------------+
computer's turn :
+-----------------------------+
|         |         |         |
|    X    |    2    |    O    |
|         |         |         |
+-----------------------------+
|         |         |         |
|    O    |    X    |    6    |
|         |         |         |
+-----------------------------+
|         |         |         |
|    7    |    8    |    X    |
|         |         |         |
+-----------------------------+
winner is  X
```