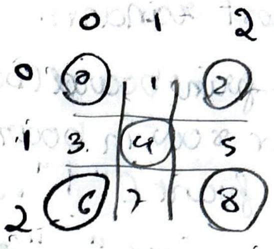
Tic-Tac-Toe game



1) A function to define the board (3×3)
```
def board-def
    print ("|" join.row( ))
    print ("-" * 9)
```

2) We take three functions
First function checks if all the elements in the row are same for i in range(3)
```
    if (board [i][0] == board [i][i] == board [i][2] != " ")
        return board[i][0]
```

Second function checks if all the elements in the column are same
```
    if (board (0)(i) == board [i](i) == board [2][i] != " ")
        return board [0][i]
```

Third function checks if all the elements in the diagonal are same
```
                                          → return board(0)(0)
if (board[0][0] == board [1][1] == board [2][2] != " " )
if (board [0][2] == board [1][1] == board [2][0] != " " )
return board [0][2]
```

3) Next function checks if the spaces are full
```
    if (cell != "    ")
        print ("All the cells are full. Try again ")
```

4) Next is the main function
```
for row in range(3)     for col in range(3)
row = int(input ("Enter the row: "))
col = int(input ("Enter the col: "))
ch = "X"
if (board (row)(col) == "  ")
    board (row) (col) : ch
else
    print ("Get computer move")
    row, col = move (board)
```

6) def-check winner (board)
This function will return the winner of the game

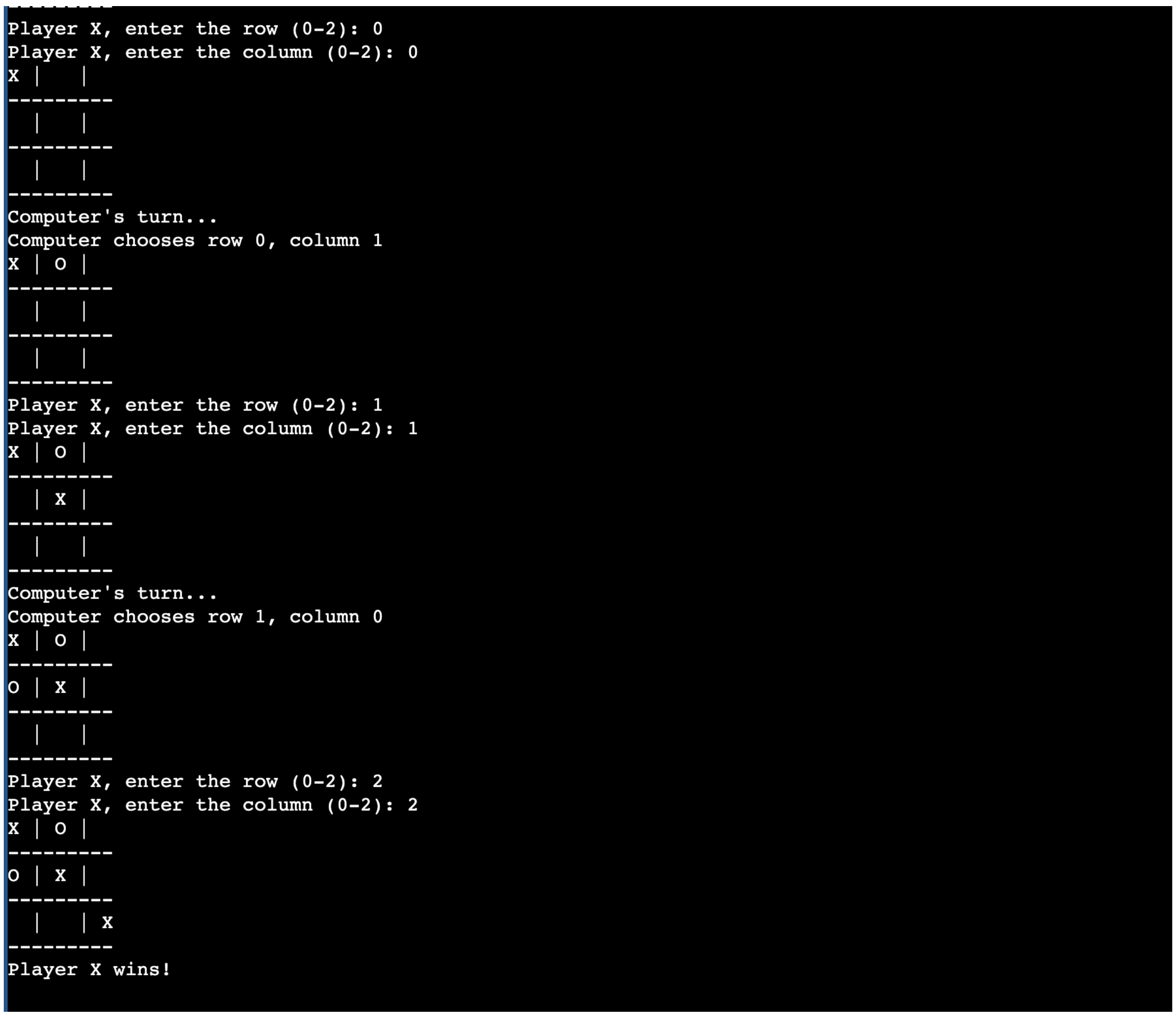
Refer Algorithm

5) In the final function move we generate a random choice for empty cells

```
import random
def-print board(board)
  for row in board:
    print(" | ".join(row))
    print("-"*9)
def-check-winner(board):
  for i in range(3):
    if (board[i][0]== board[i][1] == board[i][2] != " "):
      return board[i][0]
    if (board[0][i]== board[1][i] == board[2][i] != " "):
      return board[0][i]
    if (board[0][0] == board[1][1] == board[2][2] != " "):
      return board[0][0]
    if (board[0][2]== board[1][1] == board[2][0] != " "):
      return board[0][2]
  return None

def-is full(board):
  return all(cell != " " for row in board for cell in row)
def get-computer_move(board)
  empty-cells [(i,j) for i in range(3) for j in range(3)
  if board[i][j]==" "]
  return random.choice(empty-cells)
def tic-tac.toe():
  board=[[" " for-in range(3)] for - in range(3)]
  current-player= "X"
  computer-player= "O"
  while True:
    print-board(board)
    if current-player:"X":
      row= int(input("Player X enter the row (0-2): "))
      col = int(input("Player x enter the col (0-2): "))
```

```
else :
    print ("Computer's turn")
    row, col : get_computer_move(board)
    print (f"Computer chooses row {row}, column {col}")

if board[row][col] == " ":
    board[row][col] = current_player
else :
    print ("Cell is already taken! Try again")
    continue

winner : check_winner(board)
if winner :
    print_board (board)
    print (f"Player {winner} wins!")
    break

if is_full(board):
    print_board (board)
    print ("It's a tie!")
    break

current_player : computer_player if current_player == "X"
 else "X"

if __name__ == "__main__":
  tic_tac_toe ( \
```

[illegible] 24/9



```
---------
Player X, enter the row (0-2): 0
Player X, enter the column (0-2): 0
X |   |
---------
  |   |
---------
  |   |
---------
Computer's turn...
Computer chooses row 0, column 1
X | O |
---------
  |   |
---------
  |   |
---------
Player X, enter the row (0-2): 1
Player X, enter the column (0-2): 1
X | O |
---------
  | X |
---------
  |   |
---------
Computer's turn...
Computer chooses row 1, column 0
X | O |
---------
O | X |
---------
  |   |
---------
Player X, enter the row (0-2): 2
Player X, enter the column (0-2): 2
X | O |
---------
O | X |
---------
  |   | X
---------
Player X wins!
```