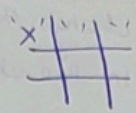


LAB-1 (24/9/24)



First take all inputs
as empty

Player = 'x'
Player = 'o'

Functions → win()
→ draw()
→ print()
→ main()

Algorithm

```
win ()
{
    if (board[0][0] == board[0][1] == board[0][2])
        return true
    if (board[1][0] == board[1][1] == board[1][2]) return true
    if (board[2][0] == board[2][1] == board[2][2]) return true
    if (board[0][0] == board[1][1] == board[2][2]) return true
    if (board[0][2] == board[1][1] == board[2][0]) return true
    if (board[0][0] == board[1][0] == board[2][0]) return true
    if (board[0][1] == board[1][1] == board[2][1]) return true
    if (board[0][2] == board[1][2] == board[2][2]) return true
    return false
}
```

horizontal checks
diagonal
vertical

```
print ()
{
    for i → 0 to rows-1
        for j → 0 to cols-1
            printf(board[i][j])
}
```

```
draw ()
{
    return true if the board is filled completely
}
```

draw → break
win → break
else invalid move

```
main ()
{
    for i → 0 to rows-1
        for j → 0 to cols-1
            board[i][j] = "" // assign empty strings
    player = 'x'
    w = false
    while (true)
    {
        Take the user input to place on the board
        if (input is out of board)
            print ("Invalid move")
            continue
        player = 'o'
    }
```



```

if (player == 'X') player = 'O'
if (player == 'O') player = 'X'
// change the player
// in the alternate manner
print()

```

```

w = win(board)
if (w) {
    print("player has won")
    break;
}

```

```

d = draw(board)

```

```

if (d) {
    print("Game is draw" "Draw!")
    break;
}

```

```

}

```

```

{
}

```

Program :

```

import random

```

```

def win(board):

```

```

    for row in board:

```

```

        if row[0] == row[1] == row[2] != " ":
            return True

```

```

    for col in range(3):

```

```

        if board[0][col] == board[1][col] == board[2][col] != " ":
            return True

```

```

        if board[0][0] == board[1][1] == board[2][2] != " ":
            return True

```

```

        if board[0][2] == board[1][1] == board[2][0] != " ":
            return True

```

```

    return False

```

```

def printBoard(board):

```

```

    print("\n".join(["|".join(row) for row in board]))

```

```

def draw(board):

```

```

    return all(cell == " " for row in board for cell in row)

```



```
def user-move(board):
```

```
while True:
```

```
try:
```

```
move = int(input("Enter your move (1-9): ")) - 1
```

```
row, col = divmod(move, 3)
```

```
if (board[row][col] == " "):
```

```
board[row][col] = "X"
```

```
break
```

```
else:
```

```
print("space is taken")
```

```
except (ValueError, IndexError):
```

```
print("Invalid Input")
```

```
def computer-move(board):
```

```
while True:
```

```
move = random.randint(0, 8)
```

```
row, col = divmod(move, 3)
```

```
if board[row][col] == " ":
```

```
board[row][col] = "O"
```

```
break
```

```
def main():
```

```
board = [[" " for _ in range(3)]]
```

```
while True:
```

```
printBoard(board)
```

```
user-move(board)
```

```
if win(board):
```

```
printBoard(board)
```

```
print("You win!")
```

```
break
```

```
if draw(board):
```

```
printBoard(board)
```

```
print("Draw!")
```

```
break
```

```
computer-move(board)
```

```
if cu in (board):
```

```
printBoard(board)
```

```
print("Computer wins!")
```

```
break
```

```

} draw(board):
    printBoard(board)
    print("Draw")
    break

```

```

if name == "__main__":
    main()

```

Output:

```

| |
| |
| |

```

Enter your move (1-9): 1

```

X | |
O | |
| |

```

Enter your move (1-9): 9

```

X | |
O | |
O | X

```

Enter your move (1-9): 5

You win!

Suka

Enter your move (1-9): 7

```

| |
O | |
X | |

```

Enter your move (1-9): 1

```

X | |
O | |
X | O

```

Enter your move (1-9): 2

```

X | X |
O | O |
X | O |

```

computer wins!