Date
Page

Algorithm Tic Tac Toe.

Step 1 : To initialize 3x3 matrix with empty cells

Step 2 : take input from user in a valid range (0-2) for rows & columns (either 'O' or 'X')

Step 3 : Check winning case after each move
for each row in board :
check for same mark ('O' or 'X')
return true (winner found)

for each column in board
check for same mark ('O' or 'X')
return true (winner found)

for each diagonal in board :
check for same mark ('O' or 'X')
return true (winner found)

else
return false.

Step 4 : If user does not input in center [1,1], place X in [2,2]. (Assuming user chooses 0)

Step 5 : If user inputs in center [1,1] traverse through matrix & input X in [0,0].

Step 6 : If user inputs in [0,2] input X in [2,0]
if user inputs in [2,0] input X in [0,2]

Date __/__/__
Page _____

step 7 : Traverse through matrix if user inputs 0
[0,1] input x in [2,1] else if
[1,0] input x in [1,2]
[2,1] input y in [0,1]
check winning function in each move

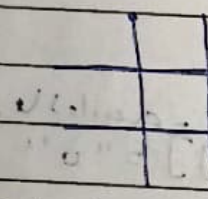
program:

```
import random

def print_board(gboard)
    for row in board:
        print("|".join(row))
        print("-" * 9)
def check_winner(board):

    for row in board
    if row.count(row[0]) == 3 and row[0] != " ":
        return row[0]


    for col in range(3):
    if board[0][col] == board[1][col] == board[2][col]
                and board[0][col] != " ":
        return board[0][col];

    if board[0][0] == board[1][1] == board[2][2]
                and board[0][0] != " ":
        return None
```

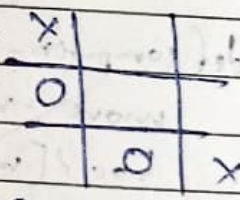
```
def is_full(board):
return all(cell != " " for row in board for cell in row)

def get_available(board):
    return [(r,c) for r in range(3) for c in range(3)
        if board[r][c] == " "]

def user_move(board):
    while True:
        move = int(input("Enter move (1-9): ")) - 1
        row, col = divmod(move, 3)
        if board[row][col] == " ":
            break
        else:
            print("cell is already taken")

def computer_move(board):
    move = random.choice(get_available_moves(board))
    board[move[0]][move[1]] = "O"

def tic_tac_toe():
    board = [[" " for _ in range(3)] for _ in range(3)]
    print_board(board)
    while True:
        user_move(board)
        print_board(board)

        if check_winner(board) == 'X':
            print("You win")
            break

        if is_full(board):
            print("draw")
            break
```

```
computer_move(board)
print_board(board)

if check_winner(board) == "O":
    print("Comp Win")
    break

if is_full(board):
    print("Draw")
    break
if
```

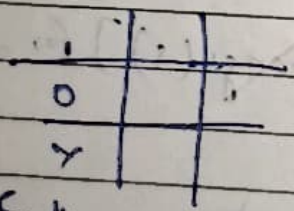
Output:



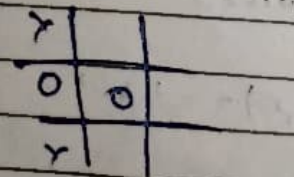
Enter Your move (1-9): 1



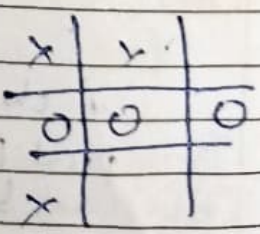
Enter your move: 5



Enter your move (1-9): 4



Enter Your move (1-9): 2



Computer wins!