Date: 24-9-24

Date __/__/__
Page _____

LAB-1

Algorithm for Tic Tac Toe

Step1: Initialize 3x3 matrix with empty cells

Step 2: Take input from user in a valid range 0 to 2 for rows and 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

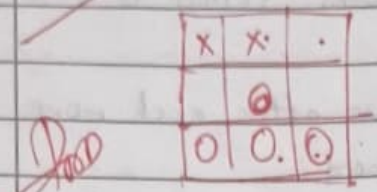
Step 4: If user does not inputs in centre [1][1], place X in [1],[2], (Assuming user chose [0,0])

Step 5: If user inputs in centre [1][1], traverse through matrix and 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 in
[0,0] input x in [2,1] else if
[1,0] input y in [1,2] else if
[2,1] input x in [0,1]
check winning function after each move.



Program:

```
import random

def print_board(board):
    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]
            c = board[2][col] and board[0][col] != " ":
            return board[0][col]

    if bord[0][0] == board[1][1] == bord[2][2]
    and bord[0][0] != " ":
```

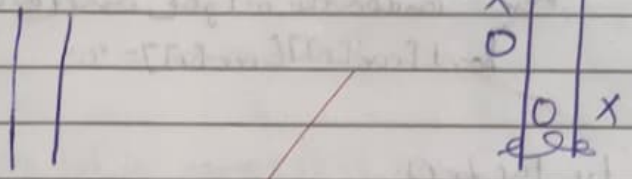
Date / /
Page

```
    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:
        try:
            move = int(input("Enter move (1-9): "))-1
            row, col = divmod(move, 3)
            if board[row][col] == " ":
                board[row][col] = "X"
                break
            else:
                print("Cell is already taken")

def computer_move(board):
    move = random.choice(get_available_move(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")
```

Date __/__/__
Page _____

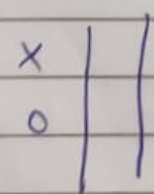
```
        break

    if is_full(board):
        print('draw')
        break

    computer_move(board)
    print_board(board)

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

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

if move == "main":
    tic_tac_toe()
```

Output:



Enter your move (1-9): 1

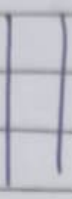
Enter your move: 5



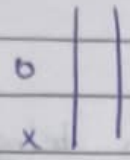
You win!!

Enter your move (1-9): 9

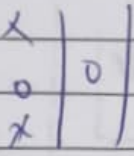
Date __/__/__
Page _____



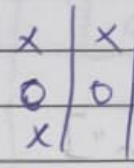
Enter your move (1-9): 7



Enter your move (1-9): 1



Enter your move (1-9): 2



Computer w/n!

24/9