## LAB 01:Tic Tac Toe Problem

DATE: 24/09/24 PAGE:

* Tic tac toe problem :-

* Algorithm :-

step 1 : create a board game
→ create 3x3 grid which has 9 cells
→ each cell can be empty or x or o

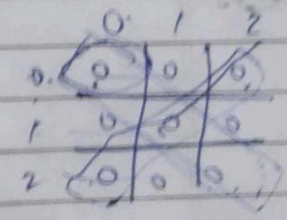
step 2 : Turn loop for the players
→ Alternate turns b/w x and o players

step 3 : Give input (place x or o respective in columns or rows where they want to place)
→ update the board after each players move.
→ After each player move, check if there is any 3 consecutive x or o, by this decide winner or draw. (vertical, diagonal, horizontal)
→ If it is winner, end the game or else it is draw when the there are no 3 consecutive x or o

step 4 : End the Game.

* Functions to print the board.
* Function to check for a win condition
* Function to get the player move
* Function to check the board.
* Function to run the game.
* ##

[0][i]




[i][0]

DATE: PAGE:

```
# def check_win(board, player)
    for row in board
        if row == [player, player, player]:
            return true

for col in range(3) ] x

def check_win(board, player)
    # check rows & columns.
for i in range(3):
    if all(board[i][j] = player for j in rang(3)):
        return true                    # check row
    if all(board[j][i) = player for i in range(3)):
        return True                    # check col

# Check diagonals
    if all [board[i][i] == player for i in range(3)):
        return True
    if all [board[i][2-i] == player for i in rang(3)):
        retur True

return flase  # No win found or draw.
```

DATE: PAGE:

code:

```
# Function to print the board
def print_board(board):
    for row in board:
        print("|".join(row))
        print("-" * 9)

# Function to check for a win condition
def check_win(board, player):
    # check rows, columns, & diagonals for a win
    for i in rang(3):
        # check rows
        if all([cell == player for cell in board[i]]):
            return True
        # check colums
        if all([board[j][i] == player for j in range(3)]):
            retur True
# check diagonals
if all([bboard[i][i] = player for i in rang(3)]) or
 all([board[i][2-i]) == player for i in range(3)]):
    return Tive
    retur Fabe
# function to check if the board is full (draw condition)
def is_draw(board):
    return all([cell != ' ' for row in board for
    cell in row]).
# Function to get the player move
def player_move(board, player):
    while True:
        try:
        move = int(input(f"player {player}, enter your
        move (1-9)"))
```

DATE: PAGE:

```
if move < 1 or move > 9:
    print ("Invalid input. Enter a number between 1 & 9")
    continue
row, col = divmod (move - 1, 3)
if board [row] [col] == ' ':
    board [row] [col] = player
    break
else:
    print ("This spot is already taken. Try again.")
except value Error
    print ("Invalid input. please enter a valid no.")

# main fn to run the game.
def tic_tac_toe():
    # initialize the board
board = [[' ' for _ in range (3)] for _ in range (3)]
# player x always starts first
current_player = 'X'
while True:
    # print the current board
    print_board (board)
# Get the player's move
    player_move (board, current_player)

# check if the current player has won
if check_win (board, current_player):
    print_board (board)
    print (f"player {current_player} win!")
    break
```

DATE: PAGE:

7)

```
# check if it's a draw
if is_draw(board):
    print_board(board)
    print("It is a draw!")
    break

# switch player
current_player = 'O' if current_player == 'X'
else 'X'

# Run the game
if __name__ == "__main__":
    tict_tact_toe()
```

Output:

player X, eenter your move (1-9)

Output:

| | | |
|---|---|---|
| X | O | X |
| O | X | X |
| X | | |

player X wins.

## Output:

```
  |   |
---------
  |   |
---------
  |   |
---------
Player X, enter your move (1-9): 1
X |   |
---------
  |   |
---------
  |   |
---------
Player O, enter your move (1-9): 2
X | O |
---------
  |   |
---------
  |   |
---------
Player X, enter your move (1-9): 3
X | O | X
---------
  |   |
---------
  |   |
---------
Player O, enter your move (1-9): 4
X | O | X
---------
O |   |
---------
  |   |
---------
Player X, enter your move (1-9): 5
X | O | X
---------
O | X |
---------
  |   |
---------
Player O, enter your move (1-9): 6
X | O | X
---------
```

```
Player X, enter your move (1-9): 3
X | O | X
---------
  |   |
---------
  |   |
---------
Player O, enter your move (1-9): 4
X | O | X
---------
O |   |
---------
  |   |
---------
Player X, enter your move (1-9): 5
X | O | X
---------
O | X |
---------
  |   |
---------
Player O, enter your move (1-9): 6
X | O | X
---------
O | X | O
---------
  |   |
---------
Player X, enter your move (1-9): 7
X | O | X
---------
O | X | O
---------
X |   |
---------
Player X wins!

=== Code Execution Successful ===-
```