Implementation Tic –Tac –Toe Game

```
import numpy as np
import random
from time import sleep
def create_board():
    return(np.array([[0, 0, 0],
                     [0, 0, 0],
                     [0, 0, 0]]))
def possibilities(board):
    l = []

    for i in range(len(board)):
        for j in range(len(board)):

            if board[i][j] == 0:
                l.append((i, j))
    return(l)
 def random_place(board, player):
    selection = possibilities(board)
    current_loc = random.choice(selection)
    board[current_loc] = player
    return(board)
def row_win(board, player):
    for x in range(len(board)):
        win = True

        for y in range(len(board)):
            if board[x, y] != player:
                win = False
                continue

        if win == True:
            return(win)
    return(win)
 def col_win(board, player):
    for x in range(len(board)):
        win = True
        for y in range(len(board)):
            if board[y][x] != player:
                win = False
                continue
        if win == True:
            return(win)
    return(win)
def diag_win(board, player):
    win = True
```

```
    y = 0
    for x in range(len(board)):
        if board[x, x] != player:
            win = False
    if win:
        return win
    win = True
    if win:
        for x in range(len(board)):
            y = len(board) - 1 - x
            if board[x, y] != player:
                win = False
def evaluate(board):
    winner = 0
    for player in [1, 2]:
        if (row_win(board, player) or
                col_win(board, player) or
                diag_win(board, player)):

            winner = player

    if np.all(board != 0) and winner == 0:
        winner = -1
    return winner
def play_game():
    board, winner, counter = create_board(), 0, 1
    print(board)
    sleep(2)
    while winner == 0:
        for player in [1, 2]:
            board = random_place(board, player)
            print("Board after " + str(counter) + " move")
            print(board)
            sleep(2)
            counter += 1
            winner = evaluate(board)
            if winner != 0:
                break
    return(winner)
 print("Winner is: " + str(play_game()))
```

Output:

```
[[0 0 0]
 [0 0 0]
 [0 0 0]]
Board after 1 move
```

```
[[0 0 0]
 [0 0 0]
 [1 0 0]]
Board after 2 move
[[0 0 2]
 [0 0 0]
 [1 0 0]]
Board after 3 move
[[0 0 2]
 [0 0 1]
 [1 0 0]]
Board after 4 move
[[0 0 2]
 [2 0 1]
 [1 0 0]]
Board after 5 move
[[0 1 2]
 [2 0 1]
 [1 0 0]]
Board after 6 move
[[0 1 2]
 [2 0 1]
 [1 2 0]]
Board after 7 move
[[1 1 2]
 [2 0 1]
 [1 2 0]]
Board after 8 move
[[1 1 2]
 [2 0 1]
 [1 2 2]]
Board after 9 move
[[1 1 2]
 [2 1 1]
 [1 2 2]]
Winner is: -1
```

1/10/24
Tuesday

Bafna Gold

LAB-I

① Implement Tic-Tac-Toe game.

~~input~~

Algorithm:

Function Print_board(board)
  For each row IN board
  PRINT row joined by " | "
    PRINT "-" * 9
FUNCTION check_winner(board)
  // check rows
    FOR each row IN board
  IF row[0] == row[1] == row[2] AND row[0] != " "
    RETURN row[0]

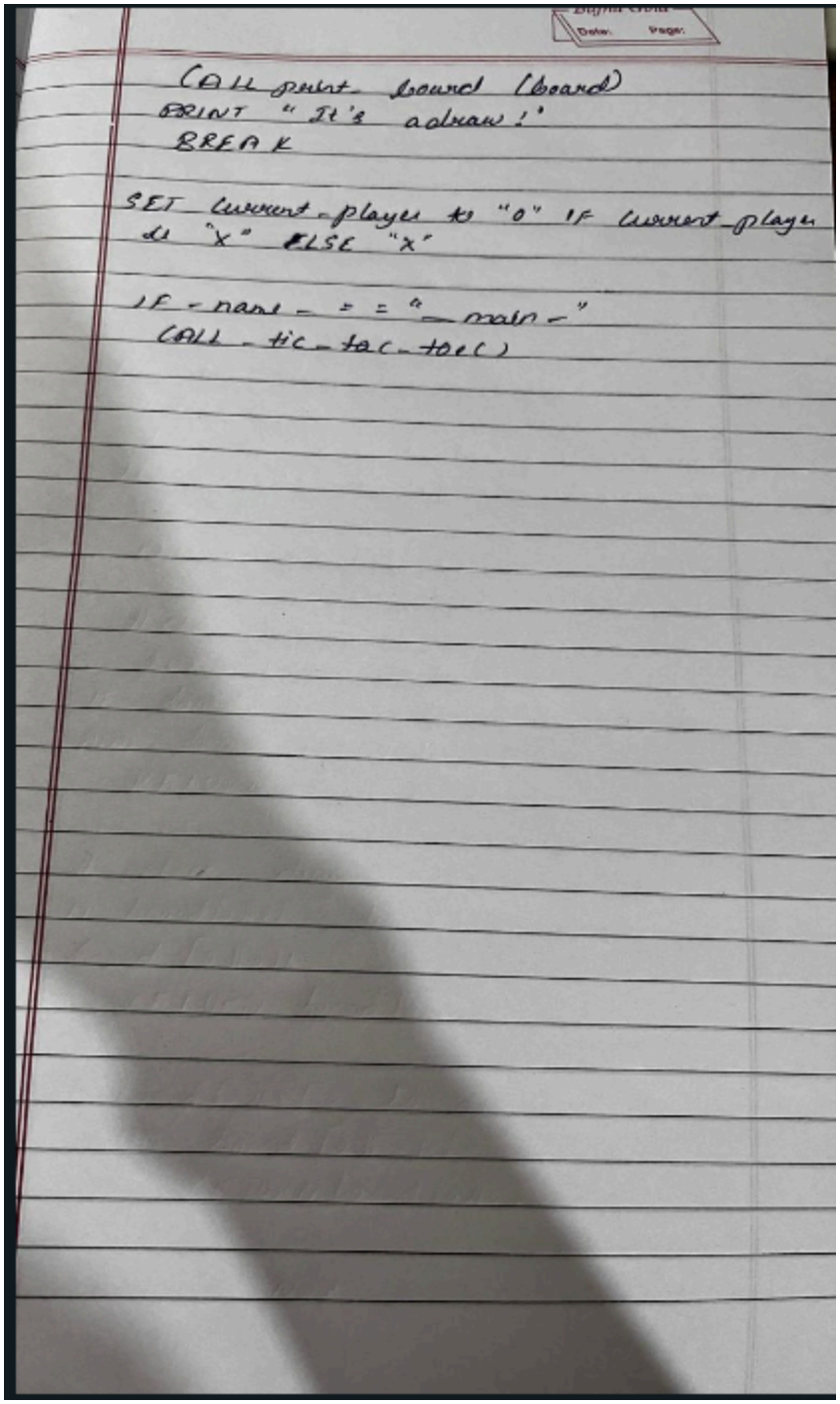
  // check columns
    FOR col From 0 TO 2
  IF board[0][col] == board[1][col] == board[2][col] AND board[0][col] != " "
    RETURN board[0][col]

  // check diagonals
  IF board[0][0] == board[1][1] == board[2][2] AND board[0][0] != " "
    RETURN board[0][0]

  IF board[0][2] == board[1][1] == board[2][0] AND board[0][2] != " "
    RETURN board[0][2]

  RETURN None

```
FUNCTION is_full (board)
RETURN True IF all cells in board are
not "" ELSE False

FUNCTION tic_tac_toe()
INITIALISE board as 3x3 board filled with ""
  SET current_player to "X"

WHILE True
  CALL Print_board (board)
PRINT "Player" + current_player + ",
  Enter your move (row and column):"

  TRY
    INPOT row, col
    IF board [row][col] != ""
    PRINT "Cell already taken. Try again"
      CONTINUE
    ENDIF
    board [row][col] = current_player
    EXCEPT (value Error, IndexError)
PRINT "Invalid input. please enter row and
column (0, 1, or 2) Seperated by space"
    CONTINUE

  SET Winner = CALL check_winner (board)
  IF winner is NOT None
  CALL Print_board (board)
PRINT "Player" + Winner + "Wins !"
    Break

  IF CALL is_full (board)
```



```
    CALL print_board (board)
    PRINT "It's a draw!"
    BREAK

SET current_player to "O" IF current_player
    is "X" ELSE "X"

IF __name__ == "__main__"
    CALL tic_tac_toe()
```