LAB 1.
24/09/24

Tic Tac Toe Game

step 1 : start

step 2 : creation of board 3×3
b[3][3] ;

step 3 : Allocate AI with O and player with X

step 4 : Take alternate input from AI and player.

step 5 : To_win()
for in in range(3):
if b[i][0] == b[i][1] == b[i][2] != ' ':
return b[i][0]
if b[0][i] == b[1][i] == b[2][i] != ' '
return b[0][i]
if b[0][0] == b[1][1] == b[2][2] != ' '
return b[0][0]
if b[0][2] == b[1][1] == b[2][0] != ' '
return b[0][2].

return -1

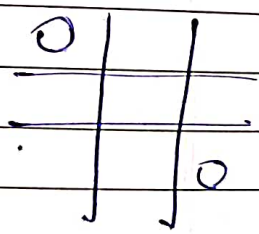
step 6 : Draw()
if (is_b_full() && either of players not won)

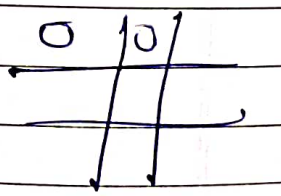
Draw has occurred
else
continuing playing

step 7 : If any of the two cells are filled and are same then return the ~~middle~~ position which is empty return position.
else call to_win.

step 8 : take input from AI and player alternatively.
if (pos == -1)
print (~~win~~ AI / player win)

step 9 : stop.





Python code :

```
def print_board(b):
    for row in b:
        print(" | ".join(row))
        print("-" * 9)

def check_winner(b):

    for i in range(3):
        if b[i][0] == b[i][1] == b[i][2] != " ":
            return b[i][0]
        if b[0][i] == b[1][i] == b[2][i] != " ":
            return b[0][i]
        if b[0][0] == b[1][1] == b[2][2] != " ":
            return b[0][0]
        if b[0][2] == b[1][1] == b[2][0] != " ":
            return b[0][2]
        return None

def is_board_full(b):
    return all(cell != " " for row in b for cell in
                row)

def get_available_moves(b):
    return [(i,j) for i in range(3) for j in range(3
            if b[i][j] == " "]

def human_move(b):
    while True:
        try:
```

```
move = int (input ("Enter your move (1-9):")) -1
if move < 0 or move > 8:
    raise ValueError
row, col = divmod (move, 3)
if b[row][col] == " ":
    return row, col
else:
    print ("Cell already taken, try again.")
except ValueError:
    print ("Invalid input, please enter a
        number between 1 and 9.")

def Computer_move (b):

for move in get_available_moves (b):
    b[move[0]][move[1]] = "O"
if check_winner (b) == "O":
    return move
b[move[0]][move[1]] = " "

for move in get_available_moves (b):
    board[move[0]][move[1]] = "X"
    if check_winner (b) == "X":
        b[move[0]][move[1]] = "O"
        return move
    b[move[0]][move[1]] = " "

return random.choice (get_available_moves(b)
```

```
def main():

    boaard = [[" " for _ in range(3)] for _ in rang(3
    print("Welcome to Tic Tac Toc! You are 'x' and
the computer is 'O'.")

    while True:
        print_b(b)
        row, col = human_move(b)
        b[row][col] = "x"

        if check_win(b) == "x":
            print_b(b)
        print("Yes You win")
        break

        if is_board_full(b):
            print_b(b)
            print("It's a tie")
            break

        print("Computer's turn")
        row, col = computer_move(b)
        b[row][col] = "O"

        if check_win(b) == "O":
            print_b(b)
            print("Computer wins!")
            break
```

```
if is_board_full(b):
    print_b(b)
    print("It's a tie!")
    break

if __name__ == "__main__":
    main()
```

output:

Welcome to tic-tac-toe! you are 'x' and the computer is 'o'.

Enter your move (1-9): 1

```
x |   |
  |   |
  |   |
o |   |
```

Enter your move (1-9): 2

```
x | x | o
  |   |
  |   |
o |   |
```

```
x | x | o
  | x |
o | o |
```

```
x | x | o
  | x |
o | o | o
```

Computer win

2/9/24