***Muhammad Kaleem***

***56614***

***Lab Task no 9***

***Q1:***

***Code:***

def evaluate(b):

for row in range(0, 3):

if b[row][0] == b[row][1] and b[row][1] == b[row][2]:

if b[row][0] == 'x':

return 10

elif b[row][0] == 'o':

return -10

for col in range(0, 3):

if b[0][col] == b[1][col] and b[1][col] == b[2][col]:

if b[0][col] == 'x':

return 10

elif b[0][col] == 'o':

return -10

if b[0][0] == b[1][1] and b[1][1] == b[2][2]:

if b[0][0] == 'x':

return 10

elif b[0][0] == 'o':

return -10

if b[0][2] == b[1][1] and b[1][1] == b[2][0]:

if b[0][2] == 'x':

return 10

elif b[0][2] == 'o':

return -10

return 0

if \_\_name\_\_ == "\_\_main\_\_":

board = [['o', '\_', 'x'],

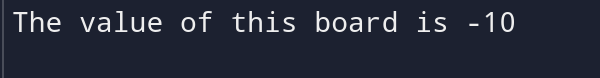
['o', 'x', 'x'],

['o', '\_', '\_']]

value = evaluate(board)

print("The value of this board is", value)

***Output:***



***Q2:***

***Code:***

def evaluate(b):

for row in range(0, 3):

if b[row][0] == b[row][1] and b[row][1] == b[row][2]:

if b[row][0] == 'x':

return 10

elif b[row][0] == 'o':

return -10

for col in range(0, 3):

if b[0][col] == b[1][col] and b[1][col] == b[2][col]:

if b[0][col] == 'x':

return 10

elif b[0][col] == 'o':

return -10

if b[0][0] == b[1][1] and b[1][1] == b[2][2]:

if b[0][0] == 'x':

return 10

elif b[0][0] == 'o':

return -10

if b[0][2] == b[1][1] and b[1][1] == b[2][0]:

if b[0][2] == 'x':

return 10

elif b[0][2] == 'o':

return -10

return 0

if \_\_name\_\_ == "\_\_main\_\_":

board = [['o', 'x', 'o'],

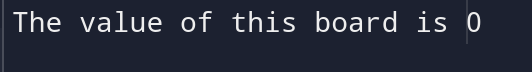
['o', 'x', 'x'],

['x', 'o', 'o']]

value = evaluate(board)

print("The value of this board is", value)

***Output:***



***Q3:***

***Code:***

def evaluate(b):

# Check rows for a win

for row in range(0, 4):

if b[row][0] == b[row][1] and b[row][1] == b[row][2]and b[row][2] == b[row][3]:

if b[row][0] == 'x':

return 10

elif b[row][0] == 'o':

return -10

# Check columns for a win

for col in range(0, 4):

if b[0][col] == b[1][col] and b[1][col] == b[2][col] and b[2][col] == b[3][col]:

if b[0][col] == 'x':

return 10

elif b[0][col] == 'o':

return -10

# Check main diagonal for a win

if b[0][0] == b[1][1] and b[1][1] == b[2][2] and b[2][2] == b[3][3]:

if b[0][0] == 'x':

return 10

elif b[0][0] == 'o':

return -10

# Check anti-diagonal for a win

if b[0][3] == b[1][2] and b[1][2] == b[2][1] and b[2][1] == b[3][0]:

if b[0][3] == 'x':

return 10

elif b[0][3] == 'o':

return -10

return 0

if \_\_name\_\_ == "\_\_main\_\_":

board = [['x', 'x', 'o', 'x'],

['x', 'x', 'x', 'o'],

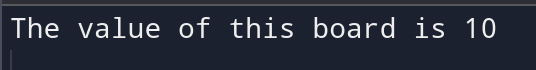
['x', 'o', 'x', 'o'],

['x', 'o', 'o', 'x']]

value = evaluate(board)

print("The value of this board is", value)

***Output:***



***Q4:***

***Code:***

def sum(a,b,c):

return a + b + c

def board(xState , zState):

zero = 'X' if xState[0] else ('O' if zState[0] else 0)

one = 'X' if xState[1] else ('O' if zState[1] else 1)

two = 'X' if xState[2] else ('O' if zState[2] else 2)

three = 'X' if xState[3] else ('O' if zState[3] else 3)

four = 'X' if xState[4] else ('O' if zState[4] else 4)

five = 'X' if xState[5] else ('O' if zState[5] else 5)

six = 'X' if xState[6] else ('O' if zState[6] else 6)

seven = 'X' if xState[7] else ('O' if zState[7] else 7)

eight = 'X' if xState[8] else ('O' if zState[8] else 8)

print(f" {zero} | {one} | {two} ")

print(f"--|---|---")

print(f" {three} | {four} | {five } ")

print(f"--|---|---")

print(f" {six} | {seven} | {eight} ")

def scenario(xState , zState):

wins = [[0,1,2],[3,4,5],[6,7,8],[0,3,6],[1,4,7],[2,5,8],[0,4,8],[2,4,6]]

for win in wins:

if(sum(xState[win[0]] , xState[win[1]] , xState[win[2]] ) == 3 ):

print(" X won the match")

return 1

if(sum(zState[win[0]] , zState[win[1]] , zState[win[2]] ) == 3 ):

print(" O won the match")

return 0

return -1

if \_\_name\_\_ == "\_\_main\_\_":

xState = [0,0,0,0,0,0,0,0,0]

zState = [0,0,0,0,0,0,0,0,0]

turn = 1

print("Welcome to Tic Tac Toe")

while(True):

board(xState , zState)

if(turn == 1):

print("X's chance ")

value = int(input("Please enter a value: "))

xState[value] = 1

else:

print("O's chance ")

value = int(input("Please enter a value: "))

zState[value] = 1

cwin = scenario(xState , zState)

if(cwin != -1):

print(" Match over ")

break

turn = 1 – turn

***Output:***

