import math

board=['-']\*9

AI='O'

YOU='X'

def print\_board(board):

for i in range(0,9,3):

print(board[i] + '|' + board[i+1] + '|' + board[i+2])

print()

def check\_winner(board,player):

winning\_combinations=[

[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 combo in winning\_combinations:

if all(board[i]==player for i in combo):

return True

return False

def is\_board\_full(board):

return all(cell!='-' for cell in board)

def minimax\_alpha\_beta(board,depth,alpha,beta,maximizing\_player):

if check\_winner(board,AI):

return 1

elif check\_winner(board,YOU):

return -1

elif is\_board\_full(board):

return 0

if maximizing\_player:

max\_eval=-math.inf

for i in range(9):

if board[i]=='-':

board[i]=AI

eval=minimax\_alpha\_beta(board,depth+1,alpha,beta,False)

board[i]='-'

max\_eval=max(max\_eval,eval)

alpha=max(alpha,eval)

if beta<=alpha:

break

return max\_eval

else:

min\_eval=math.inf

for i in range(9):

if board[i]=='-':

board[i]=YOU

eval=minimax\_alpha\_beta(board,depth+1,alpha,beta,True)

board[i]='-'

min\_eval=min(min\_eval,eval)

beta=min(beta,eval)

if beta<=alpha:

break

return min\_eval

def find\_best\_move(board):

best\_move=-1

best\_eval=-math.inf

for i in range(9):

if board[i]=='-':

board[i]=AI

eval=minimax\_alpha\_beta(board,0,-math.inf,math.inf,False)

board[i]='-'

if eval > best\_eval:

best\_eval=eval

best\_move=i

return best\_move

while True:

print\_board(board)

move=int(input("select your choice(0-8): "))

if board[move]=='-':

board[move]=YOU

if check\_winner(board,YOU):

print\_board(board)

print("You win!")

break

elif is\_board\_full(board):

print\_board(board)

print("It's a draw!")

break

ai\_move=find\_best\_move(board)

board[ai\_move]=AI

if check\_winner(board,AI):

print\_board(board)

print("AI wins!")

break

elif is\_board\_full(board):

print\_board(board)

print("It's a draw!")

break

else:

print("Cell already filled. Try again.")