**CSE-422**

**Lab Assignment -03  
*MiniMax Algorithm***

**Name: Tamanna Afroz**

**ID : 18301153**

**Section : 06**

**Task : (Alpha Beta pruning)**

print ("\n#####CSE-422-Lab1-Tamanna-Afroz-18301153#####\n")

import random as r

import math

def minimax(leaf\_nodes, branches, depth, alpha, beta, isMaximizingPlayer=True):

global itr\_count

if depth == 0:

leaf\_val = leaf\_nodes[itr\_count]

itr\_count+=1

return leaf\_val

if isMaximizingPlayer == True:

max\_val = -math.inf

for ind in range(branches):

val = minimax(leaf\_nodes, branches, depth-1, alpha, beta, False)

max\_val = max(max\_val, val)

alpha = max(alpha, val)

if beta <= alpha:

break

return max\_val

else:

min\_val = math.inf

for ind in range(branches):

val = minimax(leaf\_nodes, branches, depth-1, alpha, beta, True)

min\_val = min(min\_val, val)

beta = min(beta, val)

if beta <= alpha:

break

return min\_val

student\_id = input("Enter your student id: ")

turns = int(student\_id[0])\*2

initial\_HP = int(student\_id[-2:][::-1])

branches = int(student\_id[2])

mini,maxi = [int(x) for x in input("Minimum and Maximum value for the range of negative HP: ").split()]

x = 1

leaf\_nodes = []

while x <= branches\*\*turns:

leaf\_nodes.append(r.randrange(mini,maxi))

x+=1

print(leaf\_nodes)

itr\_count = 0

damage = minimax(leaf\_nodes,branches,turns,-math.inf,math.inf)

print("\nDepth and Branches ratio is", turns,":",branches)

print("Terminal States (leaf node values) are ",end='')

for i in leaf\_nodes:

if i != leaf\_nodes[-1]:

print(f"{i},",end='')

else:

print(f"{i}.")

print("Left life(HP) of the defender after maximum damage caused by the attacker is", initial\_HP - damage)

print("After Alpha-Beta Pruning Leaf Node Comparisons", itr\_count)