import math

import random

# My student ID is 19101147 so the program will turn the 0 into 8

student\_id = input('Enter your student ID: ')

student\_id = student\_id.replace('0', '8')

winning\_points = int(student\_id[-1] + student\_id[-2]) # exchanging the last two digits of my ID to get the winning points

min\_points = int(student\_id[4])

max\_points = int(winning\_points \* 1.5)

random\_points = [random.randint(min\_points, max\_points) for i in range(8)] # 8 random numbers in between the min and the max

# The length of random\_points is 8 i.e. 2 \*\* 3. So the height of the tree is (3 + 1).

# As the levels go from 0 to 3, we are keeping depth as it is i.e. not adding 1 to the depth var

depth = int(math.log(len(random\_points), 2))

counter = 0 # A variable to iterate over the list, random\_points

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

global counter

if depth == 0:

eval = random\_points[counter]

counter += 1

return eval

if maximizing\_player:

max\_eval = -math.inf

for child in range(2):

eval = alpha\_beta(depth - 1, alpha, beta, False)

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

alpha = max(alpha, eval)

if beta <= alpha:

'''

If child 0 is being iterated then the counter needs to be incremented otherwise,

a different parent will be accessing chlid 1. That's why if child == 0, counter += 1

'''

if child == 0:

counter += 1

break

return max\_eval

else:

min\_eval = math.inf

for child in range(2): # as every node has 2 children, the range is 2

eval = alpha\_beta(depth - 1, alpha, beta, True)

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

beta = min(beta, eval)

if beta <= alpha:

if child == 0:

counter += 1

break

return min\_eval

def task1():

points = alpha\_beta(depth, -math.inf, math.inf, True)

print('\nGenerated 8 random points between the minimum and maximum point \nlimits:', random\_points)

print('Total points to win:', winning\_points)

print('Achieved point by applying alpha-beta pruning =', points)

if points > winning\_points:

print('The winner is Optimus Prime')

else:

print('The winner is Megatron')

def task2():

global counter

shuffling\_times = int(student\_id[3]) # the fourth digit of my ID is the no of times the shuffling gonna take place

points\_list = []

for i in range(shuffling\_times):

random.shuffle(random\_points)

counter = 0

points = alpha\_beta(depth, -math.inf, math.inf, True)

points\_list.append(points)

print('\nAfter the shuffle:')

print('List of all points values from each shuffle:', points\_list)

print('The maximum value of all shuffles:', max(points\_list))

# times\_won = 0

# for i in points\_list:

# if i > winning\_points:

# times\_won += 1

times\_won = sum(i > winning\_points for i in points\_list) # One liner for counting the no of times Optimus Prime wins

print(f"Won {times\_won} times out of {shuffling\_times} number of shuffles")

task1() # function for task 1

task2() # function for task 2