import random

inp\_id = '25485465'

id = ''

for i in inp\_id:

    if i == str(0):

        id += '8'

    else:

        id += i

min\_num = int(id[4])

goal = int(id[-1] + id[-2])

max\_num = int(goal\*1.5)

suffle\_num = int(id[3])

limit = []

for i in range(8):

    limit.append(random.randint(min\_num, max\_num))

print(limit)

def min\_max(points, depth, position, alpha, beta, flag):

    if depth == 1:

        return points[position]

    if flag == True:

        max\_ = -99999

        for i in range(2):

            var = min\_max(points, depth-1,(position\*2)+i, alpha, beta, False)

            max\_ = max(max\_, var)

            alpha = max(alpha, var)

            if beta <= alpha:

                break

        return max\_

    else:

        min\_ = 99999

        for i in range(2):

            var = min\_max(points, depth-1, (position\*2)+i, alpha, beta, True)

            min\_ = min(min\_, var)

            alpha = min(alpha, var)

            if beta <= alpha:

                break

        return min\_

prime\_point = min\_max(limit, 4, 0, -99999, 99999, True)

print(f"Generated 8 random points between the minimum and maximum point limits: {limit}")

print(f"Total points to win: {goal}")

print(f"Achieved point by applying alpha-beta pruning = {prime\_point}")

if prime\_point >= goal:

    print("The winner is Optimus Prime")

else:

    print("The Winner is Megatron")

prime\_list\_point = []

for i in range(suffle\_num):

    limit = []

    for j in range(len(id)):

        limit.append(random.randint(min\_num, max\_num))

    prime\_point = min\_max(limit, 4, 0, -99999, 99999, True)

    prime\_list\_point.append(prime\_point)

count = 0

for k in range(len(prime\_list\_point)):

    if prime\_list\_point[k] > goal:

        count+=1

print(f"After the shuffle:")

print(f"List of all points values from each shuffle: {prime\_list\_point}")

print(f"The maximum value of all shuffles: {max(prime\_list\_point)}")

print(f"Won {count} times out of {len(prime\_list\_point)} number of shuffles")