import random

def alphaBetaPruning(depth, nodeidx, maxP, values, alpha, beta):

    if depth == 3:

        return values[nodeidx]

    if maxP:

        best = MIN

        for i in range(0, 2):

            val = alphaBetaPruning(depth + 1, nodeidx \* 2 + i, False, values, alpha, beta)

            best = max(best, val)

            alpha = max(alpha, best)

            if beta <= alpha:

                break

        return best

    else:

        best = MAX

        for i in range(0, 2):

            val = alphaBetaPruning(depth + 1, nodeidx \* 2 + i, True, values, alpha, beta)

            best = min(best, val)

            beta = min(beta, best)

            if beta <= alpha:

                break

        return best

##=================== Main Code =====================

x = input()

shuffle = int(x[3])

id = x[-2] + x[-1]

minV = int(x[4])

maxV = int(int(id[::-1]) \* 1.25)

to\_win = int(id[::-1])

values = []

for i in range(8):

    values.append(random.randint(minV, maxV))

MAX, MIN = 1000, -1000

print('Generated 8 random points between the minimum and maximum point limits:', values)

print('Total points to win:', to\_win)

res = alphaBetaPruning(0, 0, True, values, MIN, MAX)

print("Achieved point by applying alpha-beta pruning =", res)

if res >= to\_win:

    print("The winner is Optimus Prime")

print()

print()

tracker = []

for i in range(shuffle):

    random.shuffle(values)

    val = alphaBetaPruning(0, 0, True, values, MIN, MAX)

    tracker.append(val)

count = 0

for i in tracker:

    if i >= to\_win:

        count += 1

print('After the shuffle:')

print('List of all points values from each shuffles:', tracker)

print('The maximum value of all shuffles:', max(tracker))

print('Won', count, 'times out of 8 number of shuffles.')