import math  
import random as r  
  
std\_id = input("Enter student ID : \n")  
total\_turns = int(std\_id[0]) \* 2  
  
ini\_HP = int(std\_id[-2:][::-1])  
  
minRange, maxRange = [int(char) for char in input().split()]  
# print(minRange, maxRange)  
  
total\_branch = int(std\_id[2])  
  
leaves = []  
for i in range(0, total\_turns \*\* total\_branch + 1):  
 leaves.append(r.randrange(minRange, maxRange + 1))  
  
  
  
# print(leaves)  
print("1. Depth and Branch ratio is", total\_turns, ":", total\_branch)  
print("2. Terminal States(Leaf Nodes) are ", \*leaves)  
  
  
cnt = 0  
  
'''leaves = [18,13,5,12,10,5,13,7,17,8,6,8,5,11,13,18]  
print(leaves)'''  
  
  
def alpha\_beta\_pruning(depth, pos, mxPlayer, leaf, alp, beta):  
 global cnt  
 global total\_turns  
 global total\_branch  
  
 l = len(leaves)  
  
  
 if depth == total\_turns:  
 return leaf[pos]  
  
 if mxPlayer:  
  
 bestVal = -math.inf  
  
 for j in range(0, total\_branch):  
  
 val = alpha\_beta\_pruning(depth + 1, pos \* total\_branch + j, False, leaf, alp, beta)  
 bestVal = max(bestVal, val)  
 alp = max(alp, bestVal)  
  
 if beta <= alp:  
 corner\_case = int((l - 1) / (total\_turns - depth))  
 cur\_pos = pos \* total\_branch + j  
 if corner\_case - 1 != cur\_pos:  
 cnt += (total\_branch - (cur\_pos % total\_branch))-1  
 break  
  
 return bestVal  
  
 else:  
 bestVal = math.inf  
  
 for j in range(0, total\_branch):  
  
 val = alpha\_beta\_pruning(depth + 1, pos \* total\_branch + j, True, leaf, alp, beta)  
 bestVal = min(bestVal, val)  
 beta = min(beta, bestVal)  
  
  
 if beta <= alp:  
 corner\_case = int((l - 1) / (total\_turns - depth))  
 cur\_pos = pos \* total\_branch + j  
 if corner\_case - 1 != cur\_pos:  
 cnt += (total\_branch - (cur\_pos % total\_branch)) - 1  
 break  
 return bestVal  
  
  
algo = alpha\_beta\_pruning(0, 0, True, leaves, -math.inf, math.inf)  
print("3. Left life(HP) of the defender after maximum damage caused by the attacker is", ini\_HP - algo)  
print("4. After Alpha-Beta Pruning Leaf Node Comparisons ", (len(leaves)-cnt))