import random  
  
id=(input('Enter your student id: '))  
  
min\_max=input('Minimum and Maximum value for the range of negative HP:')  
  
min\_max=min\_max.split(' ')  
low=int(min\_max[0])  
high=int(min\_max[1])  
  
total\_turns=str(int(id[0])\*2)  
bullets=id[2]  
  
print(f"1. Depth and Branches ratio is {total\_turns}:{bullets}")  
  
  
terminal=[]  
bullets=int(bullets)  
total\_turns=int(total\_turns)  
n=(bullets)\*\*(total\_turns)  
  
for i in range (n):  
 a=random.randint(low,high)  
 terminal.append(a)  
  
print(f"2, Terminal States(Leaf Nodes) are {terminal}")  
  
initial\_hp=id[-2:]  
initial\_hp=initial\_hp[::-1]  
initial\_hp=int(initial\_hp)  
  
count=0  
leaf=0  
def minimax(position, depth, alpha, beta, maximizingPlayer, count=0, leaf=0):  
 if depth==0 and count==0:  
 leaf+=1  
 count+=1  
 return (terminal[position],count,leaf)  
 if depth==0 and count!=0:  
 leaf+=1  
 position=count  
 count+=1  
 return (terminal[position],count,leaf)  
 if maximizingPlayer:  
  
 maxEval=-10000  
  
 for i in range (bullets):  
 eval=minimax(i, depth-1,alpha, beta, False,count,leaf)  
 leaf=eval[2]  
 count=eval[1]  
 eval=eval[0]  
 maxEval=max(maxEval,eval)  
 alpha= max(alpha, eval)  
 if alpha>=beta:  
 count=count+bullets-1  
 break  
 return (maxEval,count,leaf)  
  
 else:  
 minEval=+10000  
 for i in range (bullets):  
 eval=minimax(i,depth-1,alpha,beta,True,count,leaf)  
 leaf = eval[2]  
 count = eval[1]  
 eval = eval[0]  
 minEval=min(minEval,eval)  
 beta=min(beta,eval)  
 if alpha>=beta:  
 count=count+bullets-1  
 break  
 return (minEval,count,leaf)  
  
  
a=minimax(0,total\_turns,-10000,+10000,True)  
  
print(f"3. Left life(HP) of the defender after maximum damage caused by the attacker is {initial\_hp-a[0]}")  
print(f"4. After Alpha-Beta Pruning Leaf Node Comparisons {a[2]}")