*# lab Assignment 03  
# Md. Moynul Asik Moni  
# ID: 19101189  
# CSE422, section: 05*import random  
  
  
  
  
def child\_return(p,bf,depth):  
  
 arr = []  
  
 if depth ==0:  
 for i in range(1,bf+1,1):  
 arr.append(i)  
 else:  
 n = 0  
 for i in range(0,depth-1,1):  
 n += (bf)\*\*i  
  
 a = ((bf)\*\*depth)+((p-n)\*bf)  
 count = 1  
 for i in range(0,bf,1):  
 arr.append(a-bf+count)  
 count = count+1  
 return arr  
  
  
  
  
def minimax\_alpha\_beta(position,depth,branchingF,alpha,beta,maximizingPlayer,mini,maxi,arrIn,countt):  
 if depth == 0:  
 countt[0] = countt[0]+1  
 return random.randint(mini,maxi)  
  
 if maximizingPlayer:  
 maxEval = float(**'-inf'**)  
 temp\_child = child\_return(position,branchingF,arrIn[depth])  
 for i in range(0,len(temp\_child),1):  
 eval = minimax\_alpha\_beta(temp\_child[i],depth-1,branchingF,alpha,beta,False,mini,maxi,arrIn,countt)  
 maxEval = max(maxEval,eval)  
 alpha = max(alpha,eval)  
 if beta <= alpha:  
 break  
 return maxEval  
 else:  
 minEval = float(**'inf'**)  
 temp\_child = child\_return(position, branchingF, arrIn[depth])  
 for i in range(0, len(temp\_child), 1):  
 eval = minimax\_alpha\_beta(temp\_child[i], depth - 1,branchingF, alpha, beta,True, mini, maxi, arrIn,countt)  
 minEval = min(minEval,eval)  
 beta = min(beta,eval)  
 if beta <= alpha:  
 break  
 return minEval  
  
  
  
  
  
  
  
  
if \_\_name\_\_ == **'\_\_main\_\_'**:  
 with open(**'input.txt'**) as f:  
 turn = int(f.readline())  
 bf = int(f.readline())  
 max\_min = (f.readline()).split()  
 mini = int(max\_min[0])  
 maxi = int(max\_min[1])  
  
 depth = 2\*turn  
 j=depth  
 arrIn = []  
 for i in range(0,depth+1,1):  
 arrIn.append(j)  
 j=j-1  
  
  
 countt = [1]  
 countt[0] = 0  
 leaf\_nodes = bf\*\*depth  
 alpha = float(**'-inf'**)  
 beta = float(**'inf'**)  
 max\_amount = minimax\_alpha\_beta(0,depth,bf,alpha,beta,True,mini,maxi,arrIn,countt)  
  
 print(**"Depth:"**,depth)  
 print(**"Branch:"**,bf)  
 print(**"Terminal States(Leaf Nodes):"**,leaf\_nodes)  
 print(**"Maximum amount:"**,max\_amount)  
 print(**"Comparisons:"**,leaf\_nodes)  
 print(**"Comparisons:"**,countt[0])