#Name: Md Raihanul Islam Bhuiyan  
#Id: 20101239  
#Lab Section: 3  
  
#If the answer shouldn't be -1 but it is still showing -1, run the code few times again. It should give a valid result when it is supposed to.  
data=open('input.txt')  
data=data.read()  
data=data.split('\n')  
ld=data[1:]  
  
import random  
class genetic:  
  
  
 #modeling the data and calling other funtions from it  
 def model(self):  
 transactions=int(data[0])  
 chromosomes=[]  
 for n in range(0,5):  
 s = ''  
 for i in range(0,transactions):  
 c=str(random.randint(0,1))  
 s=s+c  
 chromosomes.append(s)  
 fit={}  
 valid = False  
 #iterating the loop 10000 times to check whether we can find a chromosome with fitness value=0  
 for i in range(0,10000):  
 for c in chromosomes:  
  
 self.fitness(fit,c)  
  
 if 0 in fit:  
 #checking if the chromosome with fitness value is not made up of only 0s  
 if fit[0]!='0'\*int(data[0]):  
 print(fit[0])  
 valid=True  
 break  
 else:  
 continue  
  
  
 #sorting the fit list by fitness values  
 sfit=sorted(fit.items())  
 fit={}  
 best\_fit=sfit[0:3]  
  
  
 chromosomes=self.crossover(best\_fit,chromosomes)  
  
 r=random.randint(0,1)  
 if r==1:  
 chromosomes=self.mutation(chromosomes)  
  
 if valid==False:  
 print(-1)  
  
  
 #checking how much closer to 0 the fitness is and storing it in a dictionary called fit  
 def fitness(self,fit,c):  
  
 sum=0  
 for i in range(0,len(c)):  
 if c[i]=='1':  
 detect=ld[i].split(' ')  
  
 if detect[0]=='l':  
  
 sum=sum-int(detect[1])  
 elif detect[0]=='d':  
 #print('f')  
 sum=sum+int(detect[1])  
 if sum<0:  
 sum=sum\*-1  
 fit[sum]=c  
  
 #crossing over between the most fit chromosomes randomly  
 def crossover(self,best\_fit,chromosomes):  
 c=random.randint(1,int(data[0])-1)  
  
 if len(best\_fit)==3:  
 temp1=best\_fit[0][1][0:c]  
 temp2=best\_fit[1][1][0:c]  
 temp3=best\_fit[2][1][0:c]  
  
 c1=temp1+best\_fit[1][1][c:]  
 c2=temp1+best\_fit[2][1][c:]  
 c3=temp2+best\_fit[0][1][c:]  
 c4=temp2+ best\_fit[2][1][c:]  
 c5=temp3+best\_fit[0][1][c:]  
 c6=temp3+best\_fit[1][1][c:]  
  
 chromosomes=[c1,c2,c3,c4,c5,c6]  
 elif len(best\_fit)==2:  
 temp1 = best\_fit[0][1][0:c]  
 temp2 = best\_fit[1][1][0:c]  
 c1 = temp1 + best\_fit[1][1][c:]  
 c2=temp2+best\_fit[0][1][c:]  
  
 chromosomes=[c1,c2]  
  
 return chromosomes  
  
 #mutating a random index of the chromosomes  
 def mutation(self,chromosomes):  
 m = random.randint(1, int(data[0]) - 1)  
 mchrom=[]  
  
 for c in chromosomes:  
 s=''  
 if c[m]=='1':  
 s=s+c[0:m]+'0'+c[m+1:]  
 else:  
 s=s+c[0:m]+'1'+c[m+1:]  
 mchrom.append(s)  
  
 return(mchrom)  
  
  
a=genetic()  
a.model()