import numpy as np

def Bank(arr, fitness, bins, mutation\_threshold):

gens = 0

while gens != 50:

new\_bin = []

fitlist = fitness(bins, arr)

for i in range(1, len(bins)):

x = (select(bins, min(fitlist)))[0]

y = (select(bins, min(fitlist)))[0]

c = crossover(x, y)

if round(np.random.uniform(0, 1), 2) < mutation\_threshold:

c = mutate(c)

new\_bin.append(c)

new\_fitlist = fitness(new\_bin, arr)

if 0 not in new\_fitlist:

res = -1

else:

res = new\_bin[new\_fitlist.index(0)]

if res != "0"\*len(arr):

return res

gens += 1

return -1

def fitness(bins, arr):

fitlist = []

for i in bins:

score = 0

for j in range(len(i)):

if arr[j][0] == "d":

if i[j] == "1":

score += int(arr[j][1])

if arr[j][0] == "l":

if i[j] == "1":

score -= int(arr[j][1])

fitlist.append(score)

return fitlist

def select(bins, fitness):

return np.random.choice(bins, 1, fitness)

def crossover(x, y):

div = int(np.random.randint(1, len(x) - 1))

return x[:div] + y[div:]

def mutate(child):

return child.replace('1', 'temp').replace('0', '1').replace('temp', '0')

def binaries(arr):

bins = []

for i in range(2\*\*len(arr)-1):

bins.append(str(np.binary\_repr(i, width=(len(arr)))))

return bins

f\_input = open("input.txt","r")

f\_output = open("output.txt","w")

input\_data = f\_input.read().split("\n")

n = int(input\_data[0])

arr = []

for i in range(1, n + 1):

arr.append(input\_data[i].split(" "))

mutation\_threshold = 0.3

res= str(Bank(arr, fitness, binaries(arr), mutation\_threshold))

print(res)

f\_output.write(res)

f\_input.close()

f\_output.close()