import numpy as np

def fitness(population, inputs):

sums = []

for i in population:

csum = 0

for j in range(len(i)):

if i[j] == '1':

if inputs[j].split()[0] == 'l':

csum -= int(inputs[j].split()[1])

else:

csum += int(inputs[j].split()[1])

sums.append(csum)

return np.array(sums)

def select(population, fit):

probability = []

s = np.sum(fit)

for i in fit:

probability.append(1-i/s)

return np.random.choice(population, 1, probability)[0]

def crossover(p1, p2):

index = np.random.randint(0, len(p1))

return p1[:index] + p2[index:]

def mutate(child):

index = np.random.randint(0, len(child))

if child[index] == '0':

child = child[:index] + '1' + child[index+1:]

else:

child = child[:index] + '0' + child[index+1:]

return child

def genetic\_algorithm(population, n, mutation\_threshold=0.3):

for i in range(10000):

fitness\_array = fitness(population, n)

new\_pop = []

for j in range(len(population)):

r1 = select(population, fitness\_array)

r2 = select(population, fitness\_array)

child = crossover(r1, r2)

if np.random.random() < mutation\_threshold:

child = mutate(child)

if child == '0'\*len(child):

continue

if fitness([child], n)[0] == 0:

return child

new\_pop.append(child)

population = new\_pop

return -1

def main(filename):

with open(filename, 'r') as f:

n = int(f.readline())

inputs = []

for i in range(n):

inputs.append(f.readline()[:-1])

population = []

count = 1

while count <= 2\*\*n-1:

chrom = ""

for i in range(n):

gene = str(np.random.choice(['0', '1'], 1)[0])

chrom += gene

if chrom != '0'\*n and chrom not in population:

population.append(chrom)

count += 1

print(genetic\_algorithm(population, inputs))

main('input.txt')

main('input2.txt')