import numpy as np

def fitness(population, n):

fitness\_score = []

for chromosome in population:

fit = 0

for i in range(len(chromosome)):

if chromosome[i] == 1:

action = data[i].split()[0]

amount = int(data[i].split()[1])

if action == 'd':

fit += amount

else:

fit -= amount

fitness\_score.append(abs(0 - fit))

return fitness\_score

def select(population, fit):

sample\_size = 2

multi\_selection = False # because we have to do cross mutation between different chromosomes

probability = [x / np.sum(fit) for x in fit]

index1, index2 = np.random.choice(len(population), sample\_size, multi\_selection, probability)

return population[index1], population[index2]

def crossover(x, y):

crossover\_point = np.random.randint(0, len(x))

crossover\_child = np.concatenate((x[:crossover\_point], y[crossover\_point:]))

return crossover\_child

def mutate(child):

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

child[index\_to\_mutate] = 1 if child[index\_to\_mutate] == 0 else 0

return child

def genetic\_algo(population, n, mutation\_threshold):

max\_iterations = 1000

goal\_fit = 0

for iteration in range(1, max\_iterations):

new\_population = []

fitness\_score = fitness(population, n)

for i in range(len(population)):

parent1, parent2 = select(population, fitness\_score)

child = crossover(parent1, parent2)

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

child = mutate(child)

if all(x == 0 for x in child):

new\_population.append(child)

continue

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

print(child, "found after", iteration, 'iterations')

return None

new\_population.append(child)

population = new\_population

return -1

# ------------ driver code -----------#

transactions = int(input())

data = []

for i in range(transactions):

data.append(input())

start\_population = 10

mutation\_threshold = 0.3

population = np.random.randint(0, 2, (start\_population, transactions))

'''calling the GA function'''

ans = genetic\_algo(population, transactions, mutation\_threshold)

if ans is not None: print(-1)