# -\*- coding: utf-8 -\*-

"""2\_19101652\_Muhtasim Mahmud.ipynb

Automatically generated by Colaboratory.

Original file is located at

https://colab.research.google.com/drive/170j3fR08HltJzMaXofnyjvxLGK3d2Mbg

"""

import numpy as py

import random

def mutation(c, amount):

r1 = random.randint(0, 3)

r2 = random.randint(0, (len(amount) - 1))

c[r1] = int(amount[r2])

return c

def produce\_child(a, b):

c1 = []

c2 = []

for i in range(4):

if i <= 1:

c1.append(a[i])

c2.append(b[i])

else:

c2.append(a[i])

c1.append(b[i])

child = [c1, c2]

return child

def fitness\_function(ch, amount):

match = 0

for i in range(4):

index = amount.index(int(ch[i]))

match = match + amount[index]

return match

def res(array, amount):

final\_result = ''

for k in range(len(amount)):

a = amount[k]

if int(a) in array:

final\_result = final\_result + '1'

else:

final\_result = final\_result + '0'

k = k+1

return final\_result

### taking input from file

transactions = []

with open('input.txt') as f:

lines = f.readlines()

for line in lines:

transactions.append(line.split())

numberOfDailyTransactions = int(transactions[0][0])

amount = []

for i in range(numberOfDailyTransactions):

if(transactions[i+1][0] == 'l'):

amount.append(int(transactions[i+1][1]) \* -1)

else:

amount.append(int(transactions[i+1][1]) \* +1)

n = 4

m = 1000

result = ''

population = []

for j in range(m):

arr = []

for x in range(4): ## this for loop is generating 4 random transaction from amount array

a = random.choice(amount)

a = int(a)

arr.append(a)

population.append(arr) ## random 4 ta number er array ke population array te appending

mutation\_threshold = 0.03

i = 0

while i < 10000:

updated\_population = []

for j in range(len(population)):

u = random.choice(population)

v = random.choice(population)

children = produce\_child(u, v)

ch1 = children[0]

ch2 = children[1]

check = random.random()

if check < mutation\_threshold:

ch1 = mutation(ch1, amount)

ch2 = mutation(ch2, amount)

updated\_population.append(ch1)

updated\_population.append(ch2)

ffc1 = fitness\_function(ch1, amount)

ffc2 = fitness\_function(ch2, amount)

if ffc1 == 0:

result = res(ch1, amount)

break

if ffc2 == 0:

result = res(ch2, amount)

break

population = updated\_population

i = i + 1

if result == '':

print("-1")

else:

print(result)