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

"""08\_18101501\_MDFahimIslam

Automatically generated by Colaboratory.

Original file is located at

https://colab.research.google.com/drive/1b8ZT\_WQE0W045vjLXiWxUjnJC9LK4kgA

"""

import random as r

inputFile = open("/content/drive/MyDrive/Fall2021/CSE422/lab2/q1")

# inputFile = open("/content/drive/MyDrive/Fall2021/CSE422/lab2/q2")

n = int(inputFile.readline())

n

lines = inputFile.readlines()

lines

data = []

for line in lines:

data.append(line.strip())

len(data)

def crossOver(x,y):

chromosome\_len = len(x)

idx = r.randrange(0,chromosome\_len-1)

# print(idx)

child\_chromosome = x[:int(chromosome\_len/2)]+y[int(chromosome\_len/2):]

return child\_chromosome

def mutation(target):

child = list(target)

for i in range(0,1):

a = r.randint(0, len(target)-1)

# print(a)

b = r.randint(0, len(target)-1)

# print(b)

c = child[a]

child[a] = child[b]

child[b] = c

return ''.join(child)

def fitness\_fn(chromosome):

fitness = 0

for i in range(len(chromosome)):

if chromosome[i] == "1":

if data[i].split()[0] == "l":

fitness -= int(data[i].split()[1])

else:

fitness += int(data[i].split()[1])

return fitness

fit = []

parentPopulation = []

def calculateFitness():

global population

for elem in population:

if fitness\_fn(elem) <=400 and fitness\_fn(elem)>=-400:

fit.append([elem," ",fitness\_fn(elem)])

parentPopulation.append(elem)

def randomSelect(population):

x = r.randint(0,int((len(population)-1)/2))

y = r.randint(int((len(population)-1)/2),len(population)-1)

return population[x],population[y]

def geneticAlgo(population,n):

global fit

global s

global parentPopulation

found = False

resultChild = ""

print("Number of Population: ",len(parentPopulation))

for j in range(0,15000):

newPopulation = []

if s in parentPopulation:

parentPopulation.remove(s)

for i in range(len(parentPopulation)):

x,y = randomSelect(parentPopulation)

child = crossOver(x,y)

if s == child:

continue

if fitness\_fn(child) == 0:

resultChild = child

found = True

break

child = mutation(child)

if s == child:

continue

if fitness\_fn(child) == 0:

resultChild = child

found = True

break

else:

newPopulation.append(child)

if found:

break

population = newPopulation

if s in population:

population.remove(s)

# print("loop",j)

if found:

print(resultChild)

else:

print(-1)

"""## Start From here"""

s=""

for i in range(n):

s+="0"

global parentPopulation

population = []

i=0

# Generating distinct Populations

while len(parentPopulation) <15:

while i<=20:

chromosome = ""

for j in range(n):

chromosome += str(r.randrange(0,2))

if chromosome not in parentPopulation:

parentPopulation.append(chromosome)

i+=1

calculateFitness()

if s in parentPopulation:

parentPopulation.remove(s)

# Run Genetic Algorithm

geneticAlgo(parentPopulation,n)