f1 = open("20101539\_Umme Abira Azmary\_CSE422\_11\_Assignment02\_Summer2024\_InputFile1.txt", "r")

import random

PChromo = []

chromo = []

line1 = f1.readline().split()

N = int(line1[0])

T = int(line1[1])

hFitness = -100

hChromo = []

initial\_flag = False

for i in range(10):

for j in range(N\*T):

chromo.append(random.randint(0,1))

PChromo.append(chromo)

chromo = []

#print(PChromo)

for ksm in range(1000):

#print(ksm)

def crossover(c\_lst):

C\_bonds = []

crossChilds = []

crossPoint = random.randint(3, 6)

#print(crossPoint)

for i in range(0, len(PChromo), 2):

number1 = random.randint(0,8)

C\_bonds.append([PChromo[number1], PChromo[number1+1]])

#print(C\_bonds)

for p in C\_bonds:

ch1 = p[0][:crossPoint]

#print(ch1)

back1 = p[0][crossPoint::]

#print(back1)

ch2 = p[1][:crossPoint]

back2 = p[1][crossPoint::]

#print(back2)

child1 = ch1+back2

crossChilds.append(child1)

child2 = ch2+back1

crossChilds.append(child2)

#print(crossChilds)

return crossChilds

var = crossover(PChromo)

def CMutate(c\_lst):

mutateChild = []

for x in c\_lst:

indx = random.randint(2,6)

val = random.randint(0, 1)

c1 = x[0:indx]

c2 = x[indx+1::]

c1.append(val)

mutate = c1+c2

mutateChild.append(mutate)

#print(c\_lst)

#print(mutateChild)

return mutateChild

mChild = CMutate(var)

def fitness\_test(f\_lst):

overlap\_vals = []

consistency\_vals =[]

overlapCourses = 0

countOne = -1

steps = 1

for childs in f\_lst:

overlapCourses = 0

for vals in childs:

if int(vals) == 1:

countOne += 1

#print(steps)

if steps % 3 == 0:

if countOne >= 0:

overlapCourses += countOne

countOne = -1

steps +=1

overlap\_vals.append(overlapCourses)

# print(overlap\_vals)

consist = 0

countONe = 0

for p in f\_lst:

consist = 0

for i in range(T):

for q in range(0, len(p), T):

#print(i+q)

if p[i+q] == 1:

countONe += 1

if countONe > 1:

consist += countONe -1

elif countONe == 0:

consist += 1

countONe = 0

consistency\_vals.append(consist)

#print(consistency\_vals)

fitness = [-u-v for u, v in zip(overlap\_vals, consistency\_vals)]

return fitness

fitness = fitness\_test(mChild)

temporarycount = 0

for last in range(len(mChild)):

if fitness[last] == 0:

initial\_flag = True

hFitness = fitness[last]

hChromo = mChild[last]

elif fitness[last] > hFitness:

hFitness = fitness[last]

hChromo = mChild[last]

if initial\_flag:

break

# print(len(mChild))

# print(len(fitness))

PChromo = mChild

finalstring = ""

for word in hChromo:

finalstring += str(word)

print(finalstring)

print(hFitness)

f1 = open("20101539\_Umme Abira Azmary\_CSE422\_11\_Assignment02\_Summer2024\_InputFile2.txt", "r")

import random

line1 = f1.readline().split()

v1 = []

v2 = []

for p in line1[0]:

v1.append(int(p))

for q in line1[1]:

v2.append(int(q))

v = []

v.append(v1)

v.append(v2)

def DMutate(c\_lst):

mutateChild = []

l1 = []

l2 = []

l3 = []

indx1 = random.randint(2,3)

indx2 = random.randint(6,7)

for x in c\_lst:

f1first = x[0:indx1]

l1.append(f1first)

f1middle = x[indx1:indx2]

l2.append(f1middle)

f1rest = x[indx2::]

l3.append(f1rest)

chromo1 = l1[0]+l2[1]+l3[0]

mutateChild.append(chromo1)

chromo2 = l1[1] + l2[0] + l3[1]

mutateChild.append(chromo2)

return mutateChild

output = DMutate(v)

finaloutputstring = ""

for big in output:

for i in big:

finaloutputstring += str(i)

finaloutputstring += " "

print(finaloutputstring)