import random

with open('input.txt', 'r') as file:

n = 0

t = 0

pname = []

pdict = {}

line = file.readline()

temp = line.split()

n, t = int(temp[0]), int(temp[1])

while True:

line = file.readline()

if not line:

break

temp = line.split()

pname.append(temp[0])

pdict[temp[0]] = int(temp[1])

def generatePop(size):

n = len(size)

inpop = []

for i in range(4):

num = ''

for k in range(n):

num += str(random.randint(0, 1))

inpop.append(num)

return inpop

def fitness(pname, pdict, t, population):

fval = []

for i in population:

val = 0

k = 0

while k < len(i):

if i[k] == "1":

val += pdict[pname[k]]

k += 1

fval.append(abs(val - t))

maxi = fval.index(max(fval))

return fval, maxi

def crossover(p):

cpoint = random.randint(1, len(p[0]) - 2)

c1 = p[0][:cpoint] + p[1][cpoint:]

c2 = p[1][:cpoint] + p[0][cpoint:]

c3 = p[2][:cpoint] + p[0][cpoint:]

c4 = p[2][:cpoint] + p[1][cpoint:]

return c1, c2, c3, c4

def mutation(crossed):

for i in range(len(crossed)):

for x in range(2):

rbit = random.randint(0, len(crossed[0]) - 1)

if crossed[i][rbit] == '0':

bit = '1'

else:

bit = '0'

crossed[i] = crossed[i][:rbit] + bit + crossed[i][rbit + 1:]

return crossed

def goal(mutated):

fit, \_ = fitness(pname, pdict, t, mutated)

for i, val in enumerate(fit):

if val == 0:

return i

return False

def geneticAlgorithm():

got = False

for j in range(100):

population = generatePop(pname)

for x in range(500):

fit, ipop = fitness(pname, pdict, t, population)

population.pop(ipop)

population = list(crossover(population))

population = mutation(population)

found = goal(population)

if found == False:

continue

else:

print(pname)

print(population[found])

got = True

break

if got == True:

break

if got == False:

print(-1)

if \_\_name\_\_ == "\_\_main\_\_":

geneticAlgorithm()