import random

with open("input\_lab2.txt", "r") as f: # input file colab a

p, t = map(int, f.readline().strip().split()). #population r expected fitness

s, pl = {}, [] #score

for \_ in range(p):

n, r = f.readline().strip().split()

s[n] = int(r)

pl.append(n) #player name list a add hocche

def c(p\_list): #playerlist

s\_p, sel = [], []

for \_ in range(p):

n = random.choice(p\_list) #randomly player nicchi

s\_p.append(n)

for i in range(p):

sel.append(1 if p\_list[i] in s\_p else 0) #plyr select hole 1, otherwise 0

return sel

def fit(sel):

return sum(s[pl[i]] for i in range(p) if sel[i] == 1)

def cross(parents):

children = [] #empty list a store kortese

for i in range(0, 4, 2):

c\_idx = random.randint(0, p-1)

c\_1 = parents[i][:c\_idx] + parents[i+1][c\_idx:]

c\_2 = parents[i+1][:c\_idx] + parents[i][c\_idx:]

c\_1, c\_2 = mut(c\_1, c\_2)

children.append(c\_1) #list a children adding

children.append(c\_2)

return children

def mut(c\_1, c\_2):

c\_1[random.randint(0, p-1)] = random.randint(0, 1)

c\_2[random.randint(0, p-1)] = random.randint(0, 1)

return c\_1, c\_2

def r\_c(p\_list):

def note(lst):

return ''.join(map(str, lst))

parents\_list = [c(p\_list) for \_ in range(4)]

for \_ in range(1000): #run loop for 1000

for parent in parents\_list:

if fit(parent) == t: #parents fitness check

return f'{pl}\n{note(parent)}'

else:

parents\_list = cross(parents\_list) #target not reached, crossover for new children

return f'{pl}\n-1'

print(r\_c(pl))