#!------------------------------------ !#

"""

Name : Tanjim Reza

Student ID : 20101065

Course : CSE422 - Artificial Intelligence

Section : 07

Lab Assignment: 2

"""

#!------------------------------------ !#

import time

import random

from pprint import pprint

# start\_time = time.time()

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

data = f.read()

data = data.split('\n')

batsman = int(data[0].split(',')[0].split(' ')[0])

target = int(data[0].split(',')[0].split(' ')[1])

data.pop(0)

runs = [ int(i.split(' ')[1]) for i in data ]

batsman\_names = [ i.split(' ')[0] for i in data ]

def fitness(model, target):

score = 0

for i in range(len(model)):

if model[i] == 1:

score += runs[i]

# print("(f) M\_Score:",score)

fit = score - target

if fit == 0:

return 20101065

else:

return abs(fit)

def target\_found(model):

if fitness(model, target) == 20101065:

return True

else:

return False

def mutate(model):

index = random.randint(0, len(model)-1)

model[index] = 1 - model[index]

# print("Mutation Point:", index)

return model

def crossover(model1, model2):

index = random.randint(0, len(model1)-1)

new\_model\_one = model1[:index] + model2[index:]

new\_model\_two = model2[:index] + model1[index:]

return new\_model\_one, new\_model\_two

def generate\_model(batsman):

new\_model = []

for i in range(batsman):

new\_model.append(

random.randint(0,1)

)

return new\_model

population\_count = 1000

population = []

iteration = 1000

for i in range(population\_count):

population.append(

generate\_model(batsman)

)

# model = [1,0,1,0,1,1,1,0]

# population.append(model)

print("Population:", len(population))

print("Iterations:", iteration)

# print("--->>> START <<<---")

# pprint(population)

DO\_WE\_ITERATE = True

SOLUTION\_FOUND = False

SOLUTION\_MODEL = None

for i in range(iteration):

for model in population:

# print("Model:",model)

fitness\_of\_current\_model = fitness(model, target)

# print("Fitness of current model:", round(fitness\_of\_current\_model,6))

#sort the population based on fitness

if target\_found(model): #! IF TARGET IS FOUND

# print("------------------")

# print(f"SOLUTION MODEL: {model}")

# print("------------------")

DO\_WE\_ITERATE = False

SOLUTION\_FOUND = True

SOLUTION\_MODEL = model

break

if DO\_WE\_ITERATE:

new\_population = []

#! SORTING THE POPULATION BASED ON FITNESS

population.sort(key=lambda x: fitness(x, target), reverse=True)

for i in range(population\_count):

offspring\_1, offspring\_2 = crossover(population[0], population[1])

#! MUTATION PROBABILITY 50%

probability = random.randint(0,1)

if probability == 1:

offspring\_1 = mutate(offspring\_1)

offspring\_2 = mutate(offspring\_2)

new\_population.append(offspring\_1)

new\_population.append(offspring\_2)

population = new\_population

if DO\_WE\_ITERATE == False:

break

# print("->>>> OUT OF LOOP <<<<<-")

if SOLUTION\_FOUND:

print("All Batsman:",batsman\_names)

print("Solution Model: ", SOLUTION\_MODEL)

#batsman name from solution model

batsman\_name = []

for i in range(len(SOLUTION\_MODEL)):

if SOLUTION\_MODEL[i] == 1:

batsman\_name.append(batsman\_names[i])

print("Selected Batsman", batsman\_name)

else:

print("-1")

# end\_time = time.time()

# print("(t) Runtime:", end\_time - start\_time)