csce350 — Data Structures and Algorithms
Fall 2020 — Project 2: Balanced Teams

Assigned: September 29
Due: October 19, 11:55pm

The purpose of this assignment is to give you some practice designing and implementing an algorithm based on the exhaustive search algorithm design strategy.

The Problem Oroku is a senior manager at a medium-sized corporation, in charge of assigning workers to various projects. He knows that some of his employees are much better workers than others; in fact, he is such a good manager that he has assigned a numerical score to each employee that measures their overall ability.

In one particular episode of the company's life, there are two projects to complete, both equally important. Thus, Oroku wants to divide his team into two equally-competent groups. That is, he needs to choose the two teams carefully to ensure that the total skill level between the two teams is equal, while still ensuring that every worker is assigned to one of the two teams.

For example, suppose these workers are available.

Name	**Ability**
Baxter	13
Lonae	37
Mephos	25
Mozar	33
Savanti	32

In that case, if Oroku forms these two teams, the total ability of each team will be equal (namely, 70).

Team A	**Team B**
Baxter	Lonae
Mephos	Mozar
Savanti	

In this project, you will create a decision support system to help Oroku allocate his employees to these kinds of equally-balanced pairs of project teams.

Your Task Your job for this project is to write a program that reads the names and skill levels of the available employees and organizes them, if possible, into two teams with equal total skill level. Specifically, you should write a C++ program that does precisely these things:

1. Read a positive integer n from standard input. This represents the number of employees.
2. Read n lines, each describing one employee. Each line will have name and a skill level, separated by a space. Each name will be a sequence of letters (i.e. with no spaces). Each skill level will be an integer (which might be negative, positive, or zero).
3. Determine, using some form of exhaustive search, how those employees can be split into two teams with equal total skill level. Because every solution has a 'twin' in which the names of the two teams are swapped, you should find only solutions in which the *first employee is assigned to Team A*.
4. Show the solutions.

- If there is a solution, show it like this, listing the names of the employees assigned to each team.

  `Team A:`

  name_1

  $\vdots$

  name_m

  `Team B:`

  name_1

  $\vdots$

  name_{n-m}

  The names within each team should be listed in the same order that they appear in the input.
- If there are multiple solutions, show all of them, separated by blank lines.
- If there are no solutions, output a line containing '`No solution.`'.

See the samples below for specific illustrations of the output format.

5. Return to Step 1 above to read and process another instance. When there are multiple instances, separate the output for each one with a line containing '`------------`'. Terminate when standard input reaches end-of-file.

Here are a few example inputs with the correct output for each one.

Sample Input 1

```
5
Baxter 13
Lonae 37
Mephos 25
Mozar 33
Savanti 32
```

Sample Output 1

```
Team A:
 Baxter
 Mephos
 Savanti
Team B:
 Lonae
 Mozar
```

Sample Input 2

```
4
Friday 13
Sarah 12
Wednesday 9
Thursday 14
4
Donkey 3
Daniel 6
Cornflake 3
Bob 6
```

Sample Output 2

```
No solution.
------------
Team A:
 Donkey
 Daniel
Team B:
 Cornflake
 Bob

Team A:
 Donkey
 Bob
Team B:
 Daniel
 Cornflake
```

Sample Input 3

```
1
Tim 99
```

Sample Output 3

```
No solution.
```

An additional much longer sample input, along with its correct corresponding output, is available on the course website.

Notes A few possibly helpful comments:

- For calibration purposes, my solution has 50 lines of code. Yours may be longer, but if you find yourself writing huge amounts of code, something has probably gone wrong.
- Keep in mind that you are expected to submit your own original work for this problem. Accessing reference material for C++ in general is fine and encouraged. However, you are strongly discouraged from conducting web searches for code specific to the corporate team allocation problem.

What to Submit You should submit, using the department's dropbox website, a single C++ source file containing all of the code for your program. We will compile this program using this command line:

```
g++ -Wall -std=c++11 yourfile.cpp
```