C++ Fundamentals: Judge Assignment 3 (JA3)

The following tasks should be submitted to the SoftUni Judge system, which will be open starting **Saturday, 23 December 2017, 10:00** (in the morning) and will close on **Friday, 12 January 2018, 23:59**. Submit your solutions here: https://judge.softuni.bg/Contests/Compete/Index/893.

After the system closes, you will be able to "Practice" on the tasks – however the "Practice" results are NOT considered in the homework evaluation.

Tasks 1 and 2 of this assignment will require submitting **compressed archive (.zip) files**, containing **a single .cpp file**. Both tasks will have a **.h** file included and you can use that file directly in your code by **#include**-ing it (the Judge system will have a copy of the file) – or you can just copy the file's contents into the **.cpp** file you submit.

For **tasks 3 and 4** of this assignment, the code for each task should be a single C++ file, the contents of which you **copy-paste** into the Judge system (like with previous judge assignments).

Please be mindful of the strict input and output requirements for each task, as well as any additional requirements on running time, used memory, etc., as the tasks are evaluated automatically and not following the requirements strictly may result in your program's output being evaluated as incorrect, even if the program's logic is mostly correct.

You can use C++03 and C++11 features in your code.

Unless explicitly stated, any integer input fits into **`int`** and any floating-point input can be stored in **`double`**.

NOTE: the tasks here are NOT ordered by difficulty level.

Task 2 – Near Cities (JA3-Task-2-Near-Cities)

You are given information about towns. Each town has a name and a position represented by a point in 2-dimensional Euclidean space (i.e. the “normal” 2D space you are used to).

The distance between two towns can be calculated by the standard formula for calculating distance between two points in 2D space. You are given a **`Vector2D.h`** file which can represent points in 2D space, can represent vectors from one point to another and can calculate the length of such a vector. That file will be present in the Judge system in the same directory as the code you submit (i.e. you can do **`#include "Vector2D.h"`**) – you can use it if you want to, or do the calculations yourself.

Write a program, which, given the names and coordinates of a set of towns, finds the two closest towns (the two towns with the smallest distance between them).

Input

On the first line of the standard input, there will be the number of towns – **N**

On each of the next lines, there will be the **name**, the **`first coordinate`** and the **`second coordinate`** of a town (i.e. **name x y**), separated by single spaces.

Output

Write a single line containing the names of the two closest towns, separated by a single "**`-`**" sign. The names should be printed in order of appearance in the input. Also, if there are multiple pairs of towns with the same distance between each other, print the one that appears first in the input. For example, if the input has the towns **Sofia**, **`Burgas`**, **`Pernik`**, entered in that order, and if the distances between all of them are the same (i.e. they form an equilateral triangle), then the output should be **`Sofia-Burgas`** (NOT Burgas-Sofia, NOT Sofia-Pernik).

Restrictions

`1 < N < 100`, The name of each town will be a sequence of English letters (**a**-**z**, **A**-**Z**), no 2 towns will have the same name or same coordinates.

Coordinates will be input as integer numbers (but calculations should be done in floating-point).

The total running time of your program should be no more than **`0.1s`**

The total memory allowed for use by your program is **5MB**

Example I/O

Example Input	Expected Output
3 Sofia 0 0 Burgas 350 0 Pernik 0 -5	Sofia-Pernik
3 A -1 -1 B -4 -1 C -3 -2	B-C

4 A 1 1 B 2 1 C 2 2 D 1 2	A-B