World Finals 2009

A. Year of More Code Jam
B. Min Perimeter
C. Doubly-sorted Grid
D. Wi-fi Towers
E. Marbles
F. Lights

Contest Analysis

Questions asked 1

Submissions

Year of More Code Jam	
5pt	Not attempted **16/17 users** correct (94%)
12pt	Not attempted **9/15 users** correct (60%)
Min Perimeter	
5pt	Not attempted **17/19 users** correct (89%)
15pt	Not attempted **4/13 users** correct (31%)
Doubly-sorted Grid	
10pt	Not attempted **16/16 users** correct (100%)
20pt	Not attempted **4/5 users** correct (80%)
Wi-fi Towers	
3pt	Not attempted **22/22 users** correct (100%)
25pt	Not attempted **9/12 users** correct (75%)
Marbles	
7pt	Not attempted **16/19 users** correct (84%)
32pt	Not attempted **2/8 users** correct (25%)
Lights	
21pt	Not attempted **2/4 users** correct (50%)
45pt	Not attempted **1/2 users** correct (50%)

Top Scores

ACRush	168
qizichao	87
wata	81
ZhukovDmitry	70
dzhulgakov	69
nika	62
Vitaliy	62
kalinov	55
halyavin	54
bmerry	50

Problem B. Min Perimeter

This contest is open for practice. You can try every problem as many times as you like, though we won't keep track of which problems you solve. Read the Quick-Start Guide to get started.

Small input 5 points	Solve B-small
Large input 15 points	Solve B-large

Problem

You will be given a set of points with integer coordinates. You are asked to compute the smallest perimeter of a triangle with distinct vertexes from this set of points.

Input

When you are ready to submit a solution, the file you will download will contain Java source code that produces the input data. You should compile and run that program (see details at the bottom of the problem statement), and it will print the input data for the problem to the standard output stream. The program is guaranteed to run within 30 seconds for the small input, and 90 seconds for the large input if you don't have other resource-intensive processes running.

The first line of the input data gives you the number of cases, **T**. **T** test cases follow. Each test case contains on the first line the integer **n**, the number of points in the set. **n** lines follow, each line containing two integer numbers $\mathbf{x_i}$, $\mathbf{y_i}$. These are the coordinates of the i-th point. There may not be more than one point at the same coordinates.

Output

For each test case, output:

```
Case #X: Y
```

where **X** is the number of the test case and Y is the minimum perimeter. Answers with a relative or absolute error of at most 10^{-9} will be considered correct. Degenerate triangles — triangles with zero area — are ok.

Limits

1 <= **T** <= 15
0 <= $\mathbf{x_i}$, $\mathbf{y_i}$ <= 10^9

Small dataset

3 <= **n** <= 10000

Large dataset

3 <= **n** <= 1000000

Sample

Input

```
public class Input {
  public static void main(String[] args) {
    // T = 1
    System.out.println(1);
    // n = 10
    System.out.println(10);
    // (x[i], y[i]) = (i, i)
    for (int i = 0; i < 10; i++)
      System.out.println("" + i + " " + i);
  }
}
```

Output

```
Case #1: 5.656854
```

How to compile and run downloaded Java program

Once you're ready to submit a solution, download the program, and run it using the following instructions to get the input file:

Linux instructions:

1. Rename the file to "Input.java" (without quotes).
2. Start a console shell (Applications -> Accessories -> Terminal) and set the current directory to the location of the file ("cd <directory name>").
3. Run

   ```
   /usr/lib/jvm/java-6-sun/bin/javac Input.java
   ```

   to compile the program.
4. Run

   ```
   /usr/lib/jvm/java-6-sun/bin/java -Xmx512M Input >test.txt
   ```

   to run the program and save the results to a file named "test.txt". Now that file contains the testcase that you need to solve.

Windows instructions:

1. Rename the file to "Input.java" (without quotes).
2. Start a console shell (Start->Run->cmd.exe and set the current directory to the location of the file ("cd <directory name>").
3. Run

   ```
   "C:\Program Files\Java\jdk1.6.0_16\bin\javac" Input.java
   ```

   to compile the program.
4. Run

   ```
   "C:\Program Files\Java\jdk1.6.0_16\bin\java" -Xmx512M Inpu
   ```

   to run the program and save the results to a file named "test.txt". Now that file contains the testcase that you need to solve.