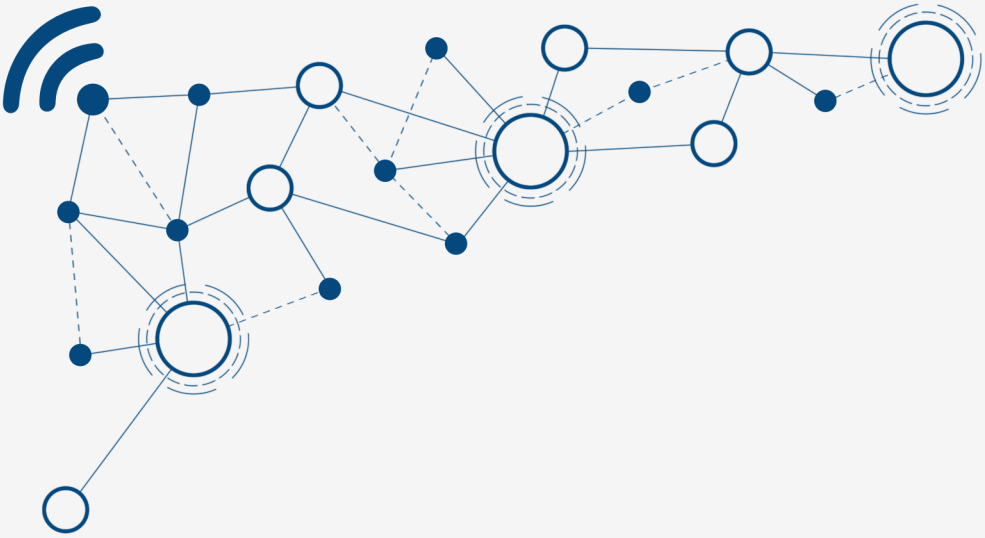
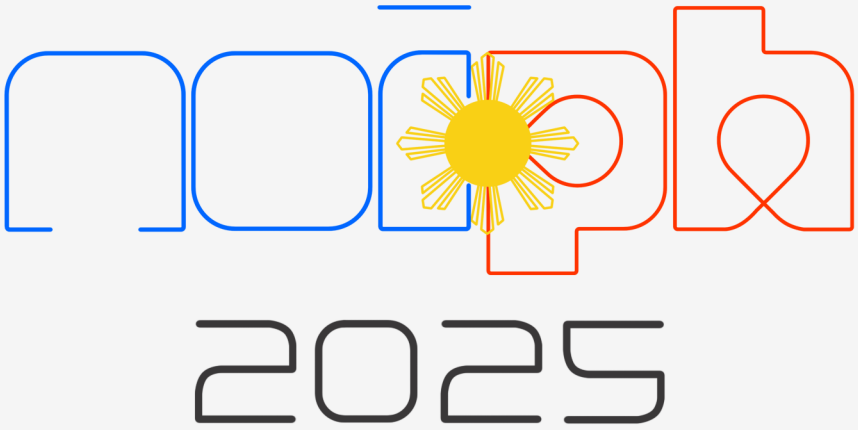
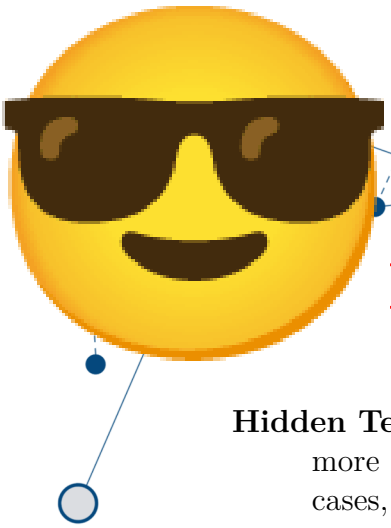





# National Olympiad in Informatics

Finals Round 1





# **Important!** Read the following:

**Hidden Test Cases.** Your solution will be checked by running it against one or more (usually several) hidden test cases. You will not have access to these cases, but a correct solution is expected to handle them correctly.

**Strict Output Format.** The output checker is **strict**. Follow these guidelines strictly:

- It is **space sensitive**. Do not output extra leading or trailing spaces. Do not output extra blank lines unless explicitly stated.
- It is **case sensitive**. So, for example, if the problem asks for the output in lowercase, follow it.
- Do not print any tabs. (No tabs will be required in the output.)
- Do not output anything else aside from what's asked for in the Output section. So, do not print things like "`Please enter t`".

Not following the output format strictly and exactly will likely result in the verdict "*Output isn't correct*".

**Use Standard I/O.** Do not read from, or write to, a file. You must read from the standard input and write to the standard output.

**Submit Code Only.** Only include **one** file when submitting: the source code (.cpp, .py, etc.) and nothing else.

**No Java Package.** For Java submissions, do not include a `package` line.

**No Weird Filenames.** Only use letters, digits and underscores in your filename. Do not use spaces or other special symbols.

**Use Fast I/O.** Many problems have large input file sizes, so use fast I/O. For example:

- In C/C++, use `scanf` and `printf`.
- In Python, use `sys.stdin.readline()`

**Flush On Interactive Problems.** On interactive problems, make sure to **flush** your output stream after printing.

- In C++, use `fflush(stdout);` or `cout << endl;`
- In Python, use `sys.stdout.flush()` or `print(flush=True)`
- For more details, including for other languages, ask a question/clarification through CMS.

Good luck and enjoy the contest! 😃

# Contents



# Notes

- Many problems have large input file sizes, so use fast I/O. For example:
  - In C/C++, use `scanf` and `printf`.
  - In Python, use `sys.stdin.readline()`
- On interactive problems, make sure to **flush** your output stream after printing.
  - In C++, use `fflush(stdout);` or `cout << endl;`
  - In Python, use `sys.stdout.flush()` or `print(flush=True)`
  - For more details, including for other languages, ask a question/clarification through CMS.

Good luck and enjoy the problems!

# Problem A
# Method of Least Squares

*In statistics, there is a well-known technique called the Method of Least Squares. The technique is used for solving a problem in statistics called "data fitting". This technique was developed in the early 1800s by the famous mathematicians Legendre, Gauss, and Laplace.*

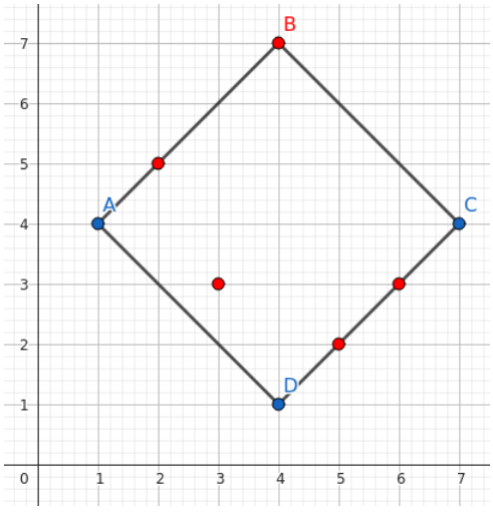
*For this problem, you are asked to recreate decades of work from these great mathematicians in the matter of a few hours.*

You are given $n$ "data points", which are literally points on a 2D plane. Your task is to find the smallest square in terms of area (also known as the *least square*) that can "fit" all of these data points. In other words, all the data points must lie either inside the square, or on the edges, or on one of the corners.

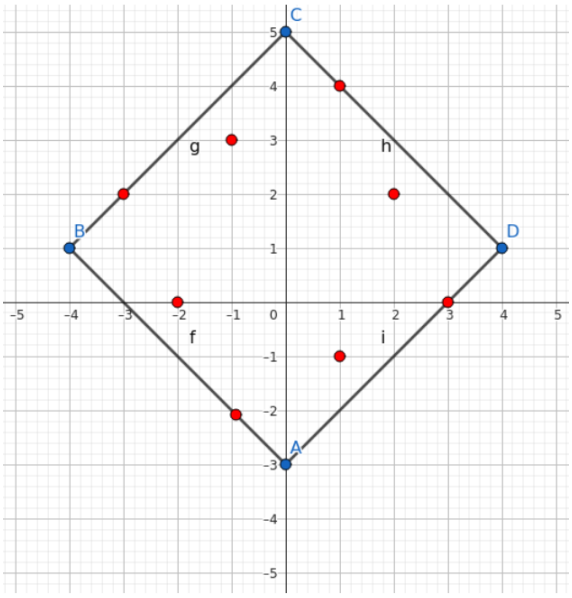
As an additional challenge, statisticians are concerned with creating "nice" models for their data sets. So, among all possible squares you can choose, you may only choose ones that satisfy *both* of the following criteria:

- It is possible to choose two of the square's corners such that those points share the same $x$ coordinate.
- It is possible to choose two of the square's corners such that those points share the same $y$ coordinate.

Here are examples of least squares, among squares that satisfy the above "niceness" condition. The data points are marked in red, and the corners of the least square are labeled $ABCD$.




Here, corners $B$ and $D$ share the same $x$-coordinate, while corners $A$ and $C$ share the same $y$-coordinate. The side length of this square is $3\sqrt{2} \approx 4.24264$, and its area is 18.




The side length of *this* square is $4\sqrt{2} \approx 5.65685$, and its area is 32.

Given our data points, please output the **minimum area** among all squares that: "fits" those data points; and is "nice".

## Input Format

The first line of input contains the single integer $n$.

Then, $n$ lines follow, each containing two space-separated integers $x$ and $y$, meaning the point $(x, y)$ is in our data set.

## Output Format

Output a single decimal value, the area of the least square that is nice.

Your answer will be accepted if it has an absolute or relative error of at most $10^{-6}$ from the judge's answer. In symbols, let $ans_{\text{you}}$ be your answer, and let $ans_{\text{judge}}$ be the judge's answer. Your answer will be accepted if

$$|ans_{\text{you}} - ans_{\text{judge}}| \leq 10^{-6}$$

In short, **make sure you print your answer to sufficiently many decimal places**. For example,

- In C++, `#include` the `<iomanip>` library, and use

  ```
  cout << fixed << setprecision(12) << ans << endl;
  ```

  to output `ans` to exactly 12 decimal places.
- In Python, use `print(f'{ans:.12f}')` to output `ans` to exactly 12 decimal places.

## Constraints

| For all subtasks |
| --- |
| $2 \leq n \leq 2 \times 10^5$<br>$\lvert x\rvert, \lvert y\rvert \leq 10^7$ for each data point.<br>The data points are unique. |

| Subtask | Points | Constraints |
| --- | --- | --- |
| 1 | **27** | $n = 2$ |
| 2 | **21** | All points lie on the positive $x$-axis. |
| 3 | **23** | Each point either:<br>lies on the positive $x$-axis, or lies on the positive $y$-axis. |
| 4 | **29** | No further constraints. |

## Sample I/O

| Input 1 | Output 1 |
| --- | --- |
| 5<br>2 5<br>3 3<br>4 7<br>5 2<br>6 3 | 18.000000 |

| Input 2 | Output 2 |
| --- | --- |
| 8<br>-3 2<br>-2 0<br>-1 -2<br>-1 3<br>1 -1<br>1 4<br>2 2<br>3 0 | 32.000000 |