

Robot World

Garima is a student of Artificial Intelligence, she is specializing in robotics. She is designing various simulations to test run her robot algorithms. The robot works in a 1-dimensional world (which has a starting point known as origin). If a robot is 5 units away from origin, then robot is said to be at the ordinate 5. Similarly if robot is 2.5 unit away from origin, then robot is said to be at the ordinate 2.5. Robots usually operate in groups. A group of robots can be described in terms of their individual ordinates. For example, if there are three robots in a group which are present at 1.3 units, 0 units and 5 units respectively away from origin, then we can say that the robot group is the ordered tuple $\langle 1.3, 0, 5 \rangle$.

One of the attributes of the robot is its power. Given a robot's ordinate we can determine the power of a robot using a given linear function called robot power function. Even though linear functions are of the form $ax + b$, but robot power functions are of the form ax and don't have any constant term. Robot's in the same group have same robot power function. For example, if robot power function is given by function $f(x) = 2x$, then robots in the group $\langle 1.3, 0, 5 \rangle$ have powers 2.6 units, 0 units and 10 units respectively. The average power of the group is given by $(2.6 + 0 + 10)/3 = 4.2$.

Given two robot groups (containing same number of robots), Garima wants to find robot power functions $f(x)$ and $g(x)$ such that $f(x) = ax$ and $g(x) = bx$ and 'a' and 'b' are positive integers and absolute difference between the average power of the two robot groups is minimized. Since 'a' and 'b' needs to be small, so $a^2 + b^2$ should also be minimized.

More mathematically, given the robot groups $\langle x_1, x_2, \dots, x_n \rangle$ and $\langle y_1, y_2, \dots, y_n \rangle$, find positive integers 'a' and 'b' such that $|(1/n) * \sum (a*x_i - b*y_i)|$ is minimized. Since, there is a possibility that there are multiple such 'a' and 'b' so Garima puts yet another constraint of minimizing $a^2 + b^2$

Input

The first line tells the number of test cases 't'.

Each test case begins with a number 'n'. 'n' corresponds to the number of robots in robot groups.

The next 'n' lines of the test case consist of two space separate real numbers. For example the i^{th} line contains real numbers ' x_i ' and ' y_i ' which represent the ordinates of one robot from each of the two robot groups.

These 'n' lines describe the two robot groups $\langle x_1, x_2, \dots, x_n \rangle$ and $\langle y_1, y_2, \dots, y_n \rangle$. It is also given that not all robots in any robot group are 0 units away from origin.

$$1 \leq t \leq 100$$

$$1 \leq n \leq 1000$$

$$0 < \sum x_i \leq 10^{14}$$

$$0 < \sum y_i \leq 10^{14}$$

x_i and y_i can have **at most 4 digits after decimal**.

Not all of x_i are zero and not all of y_i are zero.

$$\text{All } x_i \geq 0 \text{ and all } y_i \geq 0$$

#Note: x_i and y_i are **real numbers** and **NOT necessarily floating-point numbers**

You can read more about floating-point numbers here: https://en.wikipedia.org/wiki/Double-precision_floating-point_format

Output

Single line per test case.

Each line is of the form:

$$f(x) = ax, g(x) = bx$$

where 'a' and 'b' are the respective positive integers satisfying all the constraints given in the problem.

#Note: The uniqueness of 'a' and 'b' is mathematically guaranteed.

Example

Input:

3

4

7.4 0.0027

0.022 72

0.002 0.000

0.0 0

4

8.6 0.004

0.006 0.092

0.5 0

0.002 0.004

4

25 0.73

0.0062 1.4

1 0.012

0.07 0.0000

Output:

$f(x) = 720027x, g(x) = 74240x$

$f(x) = 25x, g(x) = 2277x$

$f(x) = 10710x, g(x) = 130381x$

Explanation:

For the first test case, we can observe that $|(720027 * 7.4 - 74240 * 0.0027) + (720027 * 0.022 - 74240 * 72) + (720027 * 0.002 - 74240 * 0.000) + (720027 * 0.0 - 74240 * 0)| = 0$.

Since minimum absolute value can be zero so 720027 and 74240 satisfies the minimization constraint. Also, 720027 and 74240 are the smallest such values, hence they satisfy the minimization constraint.