Concave Hull

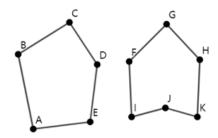| Input file:	standard input
Output file:	standard output
Time limit:	1 second
Memory limit:	1024 megabytes

A simple polygon is a closed curve in the Euclidean plane consisting of straight line segments meeting end-to-end. Two line segments meet at every endpoint, and there are no other points of intersection between the line segments.

Simple polygons can be categorized into two types: convex and concave. A convex polygon is defined as a polygon where, for any two points inside it, all points on the line segment between these two points also lie inside the polygon, either within its interior or on its boundary. A simple polygon that is not convex is called a concave polygon. As shown in the figure below, the left one is a convex polygon, while the right one is a concave polygon.

Now, given n points such that all points are distinct and no three points are collinear, your task is to select some of these n points (maybe all of them) and connect them in any order to form a **concave polygon** with a strictly positive area. You need to determine the maximum possible area of the concave polygon that can be formed.

Input

The first line contains an integer T $(1 \leq T \leq 10^4)$, indicating the number of test cases.

For each test case, the first line contains a positive integer n $(3 \leq n \leq 10^5)$, indicating the number of points.

The next n lines each contain two integers x_i, y_i $(-10^9 \leq x_i, y_i \leq 10^9)$, representing the coordinates of each point. It is guaranteed that all points are distinct, and no three points are collinear.

The sum of n over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, if it is not possible to form a concave polygon with a strictly positive area, output -1; otherwise, output a positive integer representing twice the maximum area of the concave polygon that can be formed. It can be proven that this answer is always a positive integer.

Example

standard input	standard output
2	40
6	-1
-2 0	
1 -2	
5 2	
0 4	
1 2	
3 1	
4	
0 0	
1 0	
0 1	
1 1	