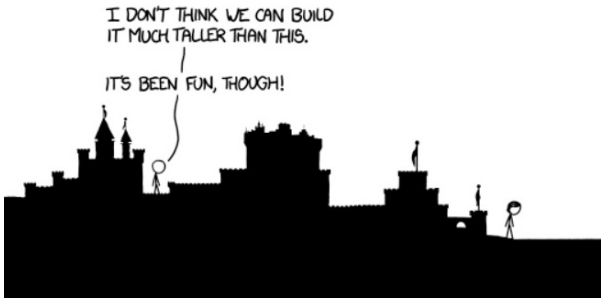
C Citadel Construction

The army wants to put up a new base in dangerous territory. The base will be surrounded by a citadel consisting of a number of straight walls. At every corner where two walls meet, there will be a watchtower. In order to enable efficient coordination in case of attack, the number of watchtowers should not exceed four.

After a detailed survey of the area, the army has concluded that not all locations are equally suitable for a watchtower: the ground needs to be firm enough to support the weight of the tower, and there should be enough visibility. They have selected a number of locations that meet the requirements.



source: xkcd.com/1190 (414)

Within these constraints, the army would like to build a base that is as large as possible. For this, they have come to you for help. Since the results of the survey are of course classified, you will not be given the possible locations directly; rather, you should write a program that can take them as input. Furthermore, they want the program to be able to handle multiple test cases in one go, so that they can hide the real data among lots of fake data.

For the computation of the area, you may assume that the watchtowers are infinitesimal in size and the walls infinitesimal in width.

Input

On the first line one positive number: the number of test cases, at most 100. After that per test case:

- one line with a single integer n ($3 \leq n \leq 1\,000$): the number of locations that are suitable for a watchtower.
- n lines, each with two space-separated integers x and y ($-10\,000 \leq x, y \leq 10\,000$): the coordinates of each location.

All locations are distinct.

Output

Per test case:

- one line with a single number: the largest possible area that the base can have. This number will be either an integer or a half-integer. If it is an integer, print that integer; if it is a half-integer, print the integer part followed by "`.5`". Trailing zeros are not allowed.

Sample in- and output

Input	Output
3 6 0 0 3 7 10 0 11 6 0 10 10 10 5 0 0 -2 -2 3 -2 0 1 0 3 10 3 1 4 1 5 9 2 6 5 3 5 8 9 7 9 3 2 3 8 4	100 12.5 31