

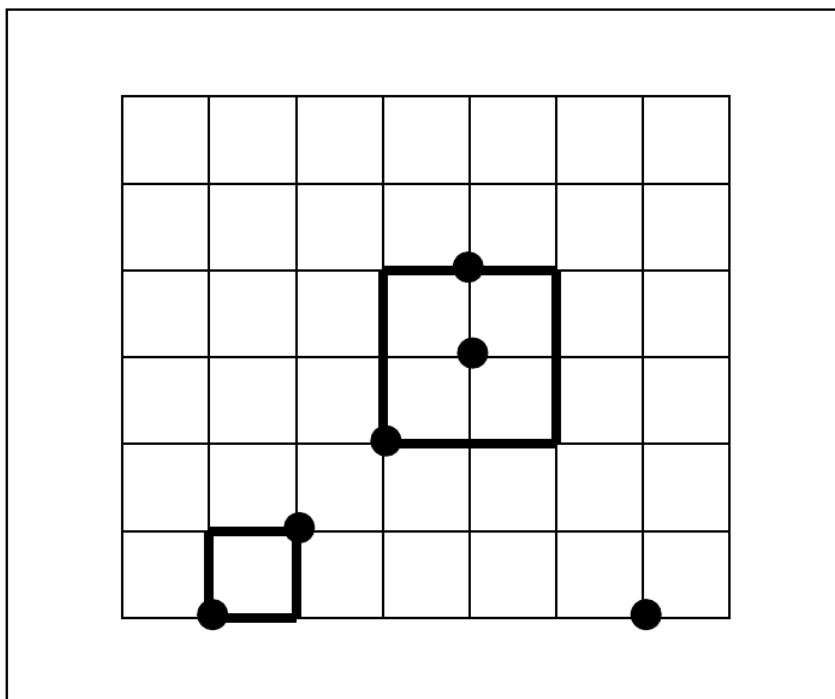
## 3282 Orchards

John wants to fence in all his trees, and happily his friend Bill has a fence building company.

Being a good friend, Bill has proposed to build the required fences for free, provided that costs are kept under control. Specifically, Bill agrees to build fences for up to three square paddocks, with sides parallel to the coordinate axes, with the condition that the side of the largest paddock(s) is as small as possible.

All paddocks must be square (have equal length sides) and paddocks must enclose all trees, considering trees as distinct points in the plane. The paddocks may intersect, or overlap on their borders, or even have 0-length sides, and a tree on a fence is considered enclosed.

A paddock may enclose only one tree, in which case its side length is 0. For example, the following diagram shows how three paddocks, of side lengths 1, 2, and 0, can be used to enclose 6 trees.



Your task is to find the minimum side of the largest paddock(s) that Bill needs to build. In the above diagram, the minimum such side is 2.

### Input

Input for this problem consists of a sequence of one or more scenarios. Each scenario is described by several lines.

- The first line contains, in order, the label of the scenario, a number,  $s$ ,  $1 \leq s \leq 100$ , followed by the number of the trees,  $n$ ,  $1 \leq n \leq 1000$ , separated by a single space.
- The first line will be followed by one or more lines containing  $2 \times n$  integers, separated by single spaces or newlines, giving, in order, the coordinates of the trees,  $x$  before  $y$ ,  $0 \leq x, y \leq 1000000$ . (Attention, for this program, input lines may contain up to 2000 characters each!)

The input will be terminated by a line consisting of two zero numbers (0), separated by a single space. This line should not be processed.

### Output

Output will be a sequence of lines, one for each input scenario. Each line will contain in order the scenario label, followed by the minimum side of the largest paddock(s) required by the problem, separated by a single space.

### Sample Input

```
1 6
1 0 2 1 3 2 4 3 4 4 6 0
2 5
1 0 2 1 3
2 5 4 6 0
3 1
1 1
4 2
1 0 5 5
0 0
```

### Sample Output

```
1 2
2 2
3 0
4 0
```