INTERNATIONAL TOURNAMENT IN INFORMATICS
Shumen, 26.11.2011

Group A

Problem A1. A STANDARD PROBLEM

In a super standard problem you are given a table of size **N**x**M**, (**N** rows and **M** columns), filled only with **0**'s and **1**'s. In such a super standard problem you are very likely to be asked: what is the biggest rectangle (a rectangle with the maximum possible area) in this table that only contains 0's?

Fortunately, nobody likes super standard problems. This is why we would modify this super standard problem into a standard one by asking for a maximum rectangle of zeroes, fit between a defined couple of rows: $\mathbf{r_1}$ (upper row) and $\mathbf{r_2}$ (lower row). Parts of the given row boundaries can take part in the solution found, but no element above the upper row or below the lower one is allowed. What is the area of the range found? Write a program **`standard`** to answer several questions of that kind for a given table.

Input

The first line of the standard input contains the table size: two space separated positive integers **N** and **M**. Each of the next **N** lines contains **M** space separated numbers (0 or 1). In the next line there is one positive integer **Q**: the number of boundary couples of rows. The next **Q** lines contain two space separated positive integers each: an upper and a lower row ($\mathbf{r_1}$, $\mathbf{r_2}$) ($1 \le \mathbf{r_1} \le \mathbf{r_2} \le \mathbf{N}$), between which you have to find a solution of the problem.

Output

The output consists of **Q** lines. In each of them you have to output one integer – the answer of the corresponding input query: the area of the biggest rectangle you can fit between the given row boundaries.

Constraints

$1 \le \mathbf{N}, \mathbf{M} \le 1000$
$1 \le \mathbf{Q} \le 10^6$

Example

Input

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

Output

```
4
6
6
```

Group A

Problem A2. CLUSTERING

Elly often spends her time with her sheep on the pasture. Doing this is not an easy task though. In addition to her sheep she also has hounds to guard the sheep from wolves and thieves. The danger for each sheep can be measured as the distance to the closest hound – the less, the better. The danger for the flock can be measured of the sum of these distances for each sheep.

If we consider the pasture as a flat surface, we can represent the sheep as *N* points in the plane. Elly wonders how to place her guarding hounds (represented as *K* points) so that the sum of minimal distances of each sheep to the closest hound was as small as possible. In other words, you are given *N* points by their *X*- and *Y*-coordinates. You should place *K* new points in such a way that the sum of the minimum distances from each of the given points to the closest of the new ones was as small as possible. Write a program `clustering` to solve this problem.

Input

On the first line of the standard input are given the space separated positive integers *N* and *K* – the number of sheep and the number of hounds, respectively. On each of the next *N* lines are given two space separated integers X_i and Y_i – the coordinates of the ith sheep.

Output

On the standard output print *K* pairs of real numbers – the coordinates of the hounds, separated with a space, formatted to the sixth digit after the decimal point. It is allowed for a hound to be placed on the same coordinates as a sheep.

Constraints

$1 \le K < N \le 1000$
$1 \le K \le 100$
$0 \le X_i, Y_i \le 10000$

Scoring

For each test case your solution will be awarded

`round(min(1, (author_score / your_score))` ^{2} `* test_score)`

points, where author_score is the result, found by the author's solution, your_score is the result of your solution, and test_score is the maximal score for the given test case.

Example

Input	**Output**
`7 2`	`1.750000 3.250000`
`1 2`	`5.000000 5.000000`
`1 4`	
`2 5`	
`3 2`	
`4 4`	
`5 6`	
`6 5`	

Clarification: Note that you are not required to print an optimal solution.

Group A

Problem A3. SUBMARINES

Fleet of N submarines is cruising in the ocean. The submarines are aligned in a row and are moving with constant speed in the same direction. The horizontal distance between each two consecutive submarines is 5 km. The positions in the fleet are numbered from *1* to N in the direction of the movement (position with number *1* is the first in the fleet and position with number N is the last in the fleet). Each submarine is moving at different depth, measured in millimeters under the ocean's surface (integer number). Each submarine can send a broadcast signal which is received only by the closest submarine which is moving behind the one sending the signal and is also deeper than it. If such submarine does not exist the signal is not received by anyone.

Explanation: The submarines are considered points – their dimensions are ignored. The closest submarine is the one that has the smallest Euclidean distance from the submarine that is sending the signal.

The Admiral of the fleet can issue two types of commands:

- „swap positions" – for that command the Admiral is giving a number of a position $1 \leq i < N$ and the submarine which is on position i must swap position with the one that is next in the fleet (on position $i+1$). The swap is executed in 0 time, each submarine is keeping it's depth and the horizontal distance between the two stays 5 km.
- „send signal " – this command is for all submarines. Each one sends a signal which is received as it was described.

Write a program **`submarines`**, which after each command of type "send signal", computes the maximum number of signals that are going to be received by a single submarine.

Input

Two positive integers N and M , separated by an interval, are given on the first row of the standard input: N is the number of the submarines in the fleet and M is the number of issued commands.

There are N positive integers on the second row – the depths on which the submarines are traveling from position *1* to N.

On each of the next M rows a single integer number is given. If the number is $i>0$, this is a "swap position" command and submarines on positions i and $i+1$ must change their positions; if the number is 0 – this is "send signal" command.

Output

For each command "send signal" your program must output a single integer on a separate row of the standard output – the computed maximum number of signals that will be received by a single submarine.

Example

Input	Output
9 3 100 300 50 1000 1100 1200 500 400 600 0 1 0	2 3

Constraints

1 mm $\leq$ depth of the submarines $\leq$ 3 000 000 mm (3 km)

Group A

Subtask 1 (20 points)

$1 < N \le 1\,000$, $1 \le M \le 100$

Subtask 2 (30 points)

$1 < N \le 1\,000\,000$, $1 \le M \le 20$

Subtask 3 (50 points)

$1 < N \le 1\,000\,000$, $1 \le M \le 100\,000$

Grading

Points for each subtask will be received only if the program solves correctly all test cases given for that particular subtask.