APS Homework

Daru is worried about a very time-consuming APS homework. As a super hacker, he develops an AI called Amadeus to help him solve his homework problems automatically.

There are N problems for this week. It takes a_i seconds for Amadeus to solve the i-th problem. Since Daru is a super hacker, he has broken into Gradescope to figure out how long it takes Gradescope to evaluate each submission. He knows that it takes b_i seconds for Gradescope autograder to evaluate and accept the correct solution for the i-th problem. (Daru is not interested in cheating by also downloading all the test cases from Gradescope since it's not cool and he knows that this would violate the academic integrity policies).

Amadeus can not work on multiple problems simultaneously. It will follow an order given by Daru to solve problems. Once Amadeus solves a problem, it submits the solution to Gradescope and continues to the next problem.

It takes no time to submit a solution and Gradescope is able to evaluate multiple solutions at the same time. The solution for the i-th problem will be accepted in exactly b_i seconds after it is submitted to Gradescope. Amadeus always produces a correct solution, so Gradescope always accepts after the first submission.

Daru wants to find an order of problems to minimize the time necessary for all problems to be accepted on Gradescope. Can you help him to figure out the optimal order?

Input

The first line contains one integer N. N is the number of problems in homework, $1 \le N \le 1000$.

The i-th line of the following N lines contains a pair of integers a_i and b_i ($1 \le a_i, b_i \le 10000$), as described above.

Output

Output one number followed by a newline. The total number of seconds that it takes Amadeus to solve, submit and get accept verdict for all the problems when the order of the problems is optimal. (Note, that there may be multiple orders that result in the same time. Since you only report on the time, the actual order is irrelevant.) The output ends with a newline.

Example 1

```
Input:
3
2 1
2 5
3 2

Output:
8
```

The optimal order is to do the 2nd problem first, then solve the 3rd problem and leave the 1st problem to solve at the end.

The solution for the 1st problem will be accepted at time 5 + 2 + 1 = 8 (Amadeus starts working on this problem at time 5).
The solution for the 2nd problem will be accepted at time 0 + 2 + 5 = 7.
The evaluation for the 3rd problem will be accepted at time 2 + 3 + 2 = 7,

Hence the total number of seconds consumed from start to end is 8.

Pocket Money

In Light Kingdom, there are $5, $10, $20, $50 and $100 notes and 5c, 10c, 20c, 50c, $1 and $2 coins. Kou's parents are going to give Kou some pocket money and your task is to figure out in how many ways that amount may be made up. For example, 20c could be made up in 4 ways: 4×5c, 10c+2×5c, 2×10c and 1×20c.

Input

The input consists of a single line. This line contains one number M (0.00 < M <= 300.00, with two decimal places), representing the amount of money Kou will receive.

Output

You should print one line consisting of M (with two decimal places and right justified in a field of width 6), followed by the number of ways in which M may be made up (right justified in a field of width 17).

Example 1

```
Input:
0.20

Output:
  0.20                 4
```

Example 2

```
Input:
2.00

Output:
  2.00               293
```

Polar Bear

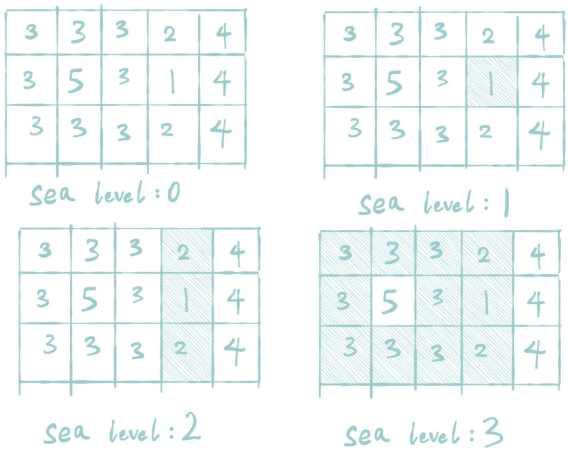
Due to global warming, the sea level is rising.

Mr. Panda is a polar bear who lives at the Bamboo Island. He is worried about flooding. Some low lying parts of Bamboo Island will be under water if the sea levels continue to rise. Mr. Panda hates swimming so it is a bad news for him.

Mr. Panda has a topographic map of the Bamboo Island. Bamboo Island has a rectangular shape, and can be divided into a n by m grid. The elevation of the each field at grid point (i,j) is A_{ij}. The soil at Bamboo Island is porous, and the water can freely flow through it. Thus, if the sea level is no less than A_{ij}, then the field at grid point (i,j) will be flooded.

Adjacent un-flooded fields (i.e., sharing common edge) create safe areas. Mr. Panda is interested in the number of distinct safe areas for a given sea level.

An example of 3x5 map is given on the right. Numbers denote the elevation of the $A_{i,j}$ as described above. Flooded fields are shaded. There will be



- 1 safe area when the sea level is 0,
- 1 safe area when the sea level is 1,
- 2 safe areas when the sea level is 2,
- 2 safe areas when the sea level is 3,
- 1 safe area when the sea level is 4 (not shown in the image),
- and no safe areas when the sea level is 5 (not shown in the image).

Input

The first line contains two numbers n and m separated by a single space, the dimensions of the Bamboo Island, where 1 <= n,m <= 1000.

Next n lines contain m integers from the range [1, 10^9] separated by single spaces, denoting the elevations of the respective fields.

Next line contains an integer T, 1 <= T <= 10^5, the number of sea levels that Mr. Panda is interested in.

The last line contains T integers t_i , separated by single spaces, such that $0 <= t_1 <= t_2 <= \ldots <= t_{T-1} <= t_T <= 10^9$. The i-th integer denotes the sea level of the i-th query.

Output

Output a single line consisting T numbers c_i separated by single spaces, where c_i is the number of safe areas when the sea level is equal to t_i. The output ends with a newline.

Example

```
Input:
3 5
3 3 3 2 4
3 5 3 1 4
3 3 3 2 4
5
1 2 3 4 5

Output:
1 2 2 1 0
```

Poppi's Rocket

The World Tree is a towering tree standing in the center of the endless Cloud Sea. The fabled paradise Elysium is said to be located at the top of the World Tree. The exterior of the World Tree is covered in thick growths of branches, while the interior is a high-tech ladder with n rungs.

Poppi can use rocket jumping to climb the ladder, but she cannot skip any rungs. Initially Poppi is on the ground. Initial strength of the rocket is set to k. As long as her jumps are strictly smaller than k, her rocket's strength remains at k. If her jump is ever equal to k, the strength decreases by 1. The rocket cannot be used to propel her to the height greater than k.

For example let the height of the rungs from the ground be 2, 7, 8, 12, 14 respectively and the initial strength of the rocket be k = 5.

Her jumps are:

1. Jumped 2 feet from the ground to the 1st rung (ground to 2), k remains 5.
2. Jumped 5 feet to the next rung (2 to 7). So, k decreases to 4.
3. Jump 1 feet to the 3rd rung (7 to 8). So, k remains 4.
4. Jump 4 feet to the 4th rung (8 to 12). So, k decreases to 3.
5. Jump 2 feet to the 5th rung (12 to 14). So, k remains 3.

Since the rockets with the greater strength cost more, Poppi asks you for help in figuring out the minimum initial strength k for the rocket so that she can reach the top rung.

Input

The first line contains 1 integer: a positive n ($0 < n \leq 10^5$) giving the number of rungs in the ladder.

The next line contains n space separated integers, $r_1, r_2, \ldots, r_n$ ($1 \leq r_1 < r_2 < \ldots < r_n \leq 10^7$), denoting the heights of the rungs from the ground.

Output

Print the minimum value of k as described above. The output ends with a newline.

Example 1

```
Input:
5
2 7 8 12 14

Output:
5
```

Ruthless War

Light Kingdom is fighting a ruthless war against Conflict Empire. As a general of Light Kingdom, you decided to attack the enemy with a linear formation of soldiers. You ordered that each soldier in the attack line should protect their two nearest neighbors in the line. These two neighbors are called pals. Note that the leftmost soldier does not have a left pal and the rightmost soldier does not have a right pal. If the either of the pals of a soldier is killed, then the next living neighbor to the left/right becomes that soldier's new pal.

As the time passes, the battle becomes much more fierce and many soldiers are being killed by light sabers, blaster pistols and magical spells. In order to keep soldier aware of their pals you need to update them about their new pals after receiving each loss report.

Input
The first line of input contains two integers `N` and `M` (`1 <= N <= M <= 100,000`) representing the number of soldiers in the attack line and the number of loss reports respectively. Soldiers are numbered from `1` to `N` according to their positions in the attack line where `1` identifies the leftmost soldier and `N` identifies the rightmost soldier. Each of the following `M` lines describes a loss report and contains two integers `L` (left) and `R` (right), meaning that soldiers from `L` to `R` were killed (`1 <= L <= R <= N`). It is guaranteed that those soldiers were alive until that moment.

Output
You should print `M` lines. In the *i*-th line should contain the new pal relationships formed after removing from the attack line the soldiers who were killed according to the *i*-th loss report. That is, for the loss report `L R` , print the first living soldier to the left of `L` and the first living soldier to the right of `R` . If there is no surviving soldier in some direction, print the character `*` instead.

Example 1

```
Input:
1 1
1 1

Output:
* *
```

Example 2

```
Input:
10 4
2 5
6 9
1 1
10 10

Output:
1 6
1 10
* 10
* *
```

Example 3

```
Input:
5 1
1 1

Output:
* 2
```