Grade Sort

You are given the midterm exam scores for all students in all of NYU campuses. The scores are natural numbers no greater than 100.

To report the midterm grades, you need to display all the scores in ascending order.

Input

The first line contains one integer N. N is the total number of grades, 1 ≤ N ≤ 4,000,000.

In the next line, there are N integers indicating the grades.

Output

Print a line with N space separated integers. These integers are the grades of the midterm sorted in ascending order.

Warning: This problem may have potentially very large input/outpu. Use fast IO operations, please.

Example 1

```
Input:
4
50 60 100 90

Output:
50 60 90 100
```

Example 1

```
Input:
6
99 92 93 92 93 91

Output:
91 92 92 93 93 99
```

Jacob’s Ladder

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 wood, 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 be 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 for the next rung (2 to 7). So, k decreases to 4.
3. Jump 1 feet for the 3rd rung (7 to 8). So, k remains 4.
4. Jump 4 feet for the 4rd rung (8 to 12). So, k decreases to 3.
5. Jump 2 feet for the 5rd 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 sot that she can reach the top rung.

Input

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

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

Output

Print the minimum value of k as described above.

Example 1

```
Input:
5
2 7 8 12 14

Output:
5
```

Polar Bear

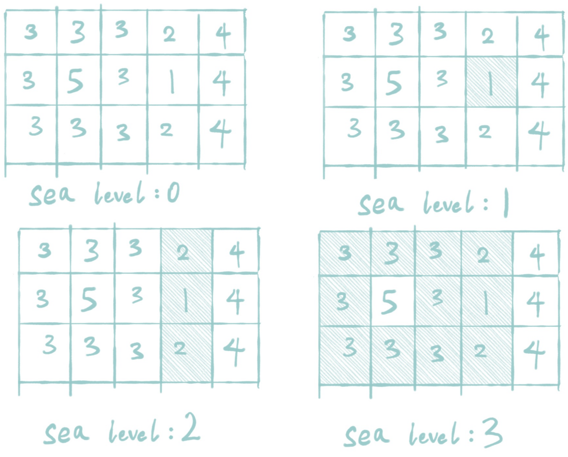
Due to global warming, the sea level is rising.

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 $0 <= t_1 <= t_2 <= . . . <= t_{T-1} <= t_T <= 10\^9$. 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

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
```

Unique Subarray

A subarray is the sequence of consecutive elements in an array. A *unique* subarray is a subarray in which all elements are unique (i.e., no repeated values). Given an array, your task is to calculate the maximum length of a *unique* subarray.

For example [4, 3, 2, 2, 1], the maximum length of of a *unique* subarray is 3. The *unique* subarray is [4,3,2]. [3,2,2,1] is a subarray of length 4, but since 2 appears twice in this subarray, it is not a *unique* subarray.

Input

The 1st line contains an integer n (1 <= n <= 10^6), specifying the length of the array.

The following line contains n integers, in range [1, 10^9], denoting the elements of the array.

Output

The maximum length of a *unique* subarray.

Example 1

```
Input:
5
4 3 2 2 1

Output:
3
```

Example 2

```
Input:
6
1 1 1 1 1 1

Output:
1
```

APS Homework

Daru is weary of the time-consuming APS homework. As a super hacker, he developed 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, $1 \leq N \leq 1000$.

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

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 (Amadeus starts at 5) + 2 + 1 = 8.
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.