A. Mashmokh and Lights

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Mashmokh works in a factory. At the end of each day he must turn off all of the lights.

The lights on the factory are indexed from 1 to n. There are n buttons in Mashmokh's room indexed from 1 to n as well. If Mashmokh pushes button with index i, then each light with index not less than i that is still turned on turns off.

Mashmokh is not very clever. So instead of pushing the first button he pushes some of the buttons randomly each night. He pushed m distinct buttons $b_1, b_2, ..., b_m$ (the buttons were pushed consecutively in the given order) this night. Now he wants to know for each light the index of the button that turned this light off. Please note that the index of button b_i is actually b_i, not i.

Please, help Mashmokh, print these indices.

Input

The first line of the input contains two space-separated integers n and m ($1 \le n, m \le 100$), the number of the factory lights and the pushed buttons respectively. The next line contains m distinct space-separated integers $b_1, b_2, ..., b_m$ ($1 \le b_i \le n$).

It is guaranteed that all lights will be turned off after pushing all buttons.

Output

Output n space-separated integers where the i-th number is index of the button that turns the i-th light off.

Examples

input
```
5 4
4 3 1 2
```
output
```
1 1 3 4 4
```

input
```
5 5
5 4 3 2 1
```
output
```
1 2 3 4 5
```

Note

In the first sample, after pressing button number 4, lights 4 and 5 are turned off and lights 1, 2 and 3 are still on. Then after pressing button number 3, light number 3 is turned off as well. Pressing button number 1 turns off lights number 1 and 2 as well so pressing button number 2 in the end has no effect. Thus button number 4 turned lights 4 and 5 off, button number 3 turned light 3 off and button number 1 turned light 1 and 2 off.

B. Mashmokh and Tokens

Bimokh is Mashmokh's boss. For the following n days he decided to pay to his workers in a new way. At the beginning of each day he will give each worker a certain amount of tokens. Then at the end of each day each worker can give some of his tokens back to get a certain amount of money. The worker can save the rest of tokens but he can't use it in any other day to get more money. If a worker gives back w tokens then he'll get $\lfloor \frac{w \cdot a}{b} \rfloor$ dollars.

Mashmokh likes the tokens however he likes money more. That's why he wants to save as many tokens as possible so that the amount of money he gets is maximal possible each day. He has n numbers $x_1, x_2, ..., x_n$. Number x_i is the number of tokens given to each worker on the i-th day. Help him calculate for each of n days the number of tokens he can save.

Input

The first line of input contains three space-separated integers n, a, b ($1 \le n \le 10^5$; $1 \le a, b \le 10^9$). The second line of input contains n space-separated integers $x_1, x_2, ..., x_n$ ($1 \le x_i \le 10^9$).

Output

Output n space-separated integers. The i-th of them is the number of tokens Mashmokh can save on the i-th day.

Examples

input
5 1 4 12 6 11 9 1
output
0 2 3 1 1

input
3 1 2 1 2 3
output
1 0 1

input
1 1 1 1
output
0

C. Mashmokh and Numbers

It's holiday. Mashmokh and his boss, Bimokh, are playing a game invented by Mashmokh.

In this game Mashmokh writes sequence of n distinct integers on the board. Then Bimokh makes several (possibly zero) moves. On the first move he removes the first and the second integer from from the board, on the second move he removes the first and the second integer of the remaining sequence from the board, and so on. Bimokh stops when the board contains less than two numbers. When Bimokh removes numbers x and y from the board, he gets $gcd(x, y)$ points. At the beginning of the game Bimokh has zero points.

Mashmokh wants to win in the game. For this reason he wants his boss to get exactly k points in total. But the guy doesn't know how choose the initial sequence in the right way.

Please, help him. Find n distinct integers $a_1, a_2, ..., a_n$ such that his boss will score exactly k points. Also Mashmokh can't memorize too huge numbers. Therefore each of these integers must be at most 10^9.

Input

The first line of input contains two space-separated integers n, k ($1 \le n \le 10^5$; $0 \le k \le 10^8$).

Output

If such sequence doesn't exist output -1 otherwise output n distinct space-separated integers $a_1, a_2, ..., a_n$ ($1 \le a_i \le 10^9$).

Examples

input
5 2
output
1 2 3 4 5

input
5 3
output
2 4 3 7 1

input
7 2
output
-1

Note

$gcd(x, y)$ is greatest common divisor of x and y.

D. Mashmokh and ACM

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Mashmokh's boss, Bimokh, didn't like Mashmokh. So he fired him. Mashmokh decided to go to university and participate in ACM instead of finding a new job. He wants to become a member of Bamokh's team. In order to join he was given some programming tasks and one week to solve them. Mashmokh is not a very experienced programmer. Actually he is not a programmer at all. So he wasn't able to solve them. That's why he asked you to help him with these tasks. One of these tasks is the following.

A sequence of l integers $b_1, b_2, ..., b_l$ $(1 \le b_1 \le b_2 \le ... \le b_l \le n)$ is called *good* if each number divides (without a remainder) by the next number in the sequence. More formally $b_i \mid b_{i+1}$ for all i $(1 \le i \le l - 1)$.

Given n and k find the number of good sequences of length k. As the answer can be rather large print it modulo 1000000007 $(10^9 + 7)$.

Input

The first line of input contains two space-separated integers n, k $(1 \le n, k \le 2000)$.

Output

Output a single integer — the number of good sequences of length k modulo 1000000007 $(10^9 + 7)$.

Examples

input
3 2
output
5

input
6 4
output
39

input
2 1
output
2

Note

In the first sample the good sequences are: $[1, 1], [2, 2], [3, 3], [1, 2], [1, 3]$.

E. Mashmokh and Reverse Operation

Mashmokh's boss, Bimokh, didn't like Mashmokh. So he fired him. Mashmokh decided to go to university and participate in ACM instead of finding a new job. He wants to become a member of Bamokh's team. In order to join he was given some programming tasks and one week to solve them. Mashmokh is not a very experienced programmer. Actually he is not a programmer at all. So he wasn't able to solve them. That's why he asked you to help him with these tasks. One of these tasks is the following.

You have an array a of length 2^n and m queries on it. The i-th query is described by an integer q_i. In order to perform the i-th query you must:

- split the array into 2^{n-q_i} parts, where each part is a subarray consisting of 2^{q_i} numbers; the j-th subarray $(1 \le j \le 2^{n-q_i})$ should contain the elements $a[(j-1) \cdot 2^{q_i} + 1], a[(j-1) \cdot 2^{q_i} + 2], \ldots, a[(j-1) \cdot 2^{q_i} + 2^{q_i}]$;
- reverse each of the subarrays;
- join them into a single array in the same order (this array becomes new array a);
- output the number of inversions in the new a.

Given initial array a and all the queries. Answer all the queries. Please, note that the changes from some query is saved for further queries.

Input

The first line of input contains a single integer n $(0 \le n \le 20)$.

The second line of input contains 2^n space-separated integers $a[1], a[2], \ldots, a[2^n]$ $(1 \le a[i] \le 10^9)$, the initial array.

The third line of input contains a single integer m $(1 \le m \le 10^6)$.

The fourth line of input contains m space-separated integers $q_1, q_2, \ldots, q_m$ $(0 \le q_i \le n)$, the queries.

Note: since the size of the input and output could be very large, don't use slow output techniques in your language. For example, do not use input and output streams (cin, cout) in C++.

Output

Output m lines. In the i-th line print the answer (the number of inversions) for the i-th query.

Examples

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

output
```
0
6
6
0
```

input
```
1
1 2
3
0 1 1
```

output
```
0
1
0
```

Note

If we reverse an array $x[1], x[2], \ldots, x[n]$ it becomes new array $y[1], y[2], \ldots, y[n]$, where $y[i] = x[n - i + 1]$ for each i.

The number of inversions of an array $x[1], x[2], \ldots, x[n]$ is the number of pairs of indices i, j such that: $i < j$ and $x[i] > x[j]$.