Educational Codeforces Round 24

A. Diplomas and Certificates

1 second, 256 megabytes

There are n students who have taken part in an olympiad. Now it's time to award the students.

Some of them will receive diplomas, some wiil get certificates, and others won't receive anything. Students with diplomas and certificates are called *winners*. But there are some rules of counting the number of diplomas and certificates. The number of certificates must be **exactly** k times greater than the number of diplomas. The number of *winners* must **not be greater than half of the number of all students** (i.e. not be greater than half of n). It's possible that there are no *winners*.

You have to identify the maximum possible number of *winners*, according to these rules. Also for this case you have to calculate the number of students with diplomas, the number of students with certificates and the number of students who are not *winners*.

Input

The first (and the only) line of input contains two integers n and k ($1 \le n, k \le 10^{12}$), where n is the number of students and k is the ratio between the number of certificates and the number of diplomas.

Output

Output three numbers: the number of students with diplomas, the number of students with certificates and the number of students who are not *winners* in case when the number of *winners* is maximum possible.

It's possible that there are no *winners*.

input
```
18 2
```
output
```
3 6 9
```

input
```
9 10
```
output
```
0 0 9
```

input
```
1000000000000 5
```
output
```
83333333333 416666666665 500000000002
```

input
```
1000000000000 499999999999
```
output
```
1 499999999999 500000000000
```

B. Permutation Game

1 second, 256 megabytes

n children are standing in a circle and playing a game. Children's numbers in clockwise order form a permutation $a_1, a_2, ..., a_n$ of length n. It is an integer sequence such that each integer from 1 to n appears exactly once in it.

The game consists of m steps. On each step the current leader with index i counts out a_i people in clockwise order, starting from the next person. The last one to be pointed at by the leader becomes the new leader.

You are given numbers $l_1, l_2, ..., l_m$ — indices of leaders in the beginning of each step. Child with number l_1 is the first leader in the game.

Write a program which will restore a possible permutation $a_1, a_2, ..., a_n$. If there are multiple solutions then print any of them. If there is no solution then print `-1`.

Input

The first line contains two integer numbers n, m ($1 \le n, m \le 100$).

The second line contains m integer numbers $l_1, l_2, ..., l_m$ ($1 \le l_i \le n$) — indices of leaders in the beginning of each step.

Output

Print such permutation of n numbers $a_1, a_2, ..., a_n$ that leaders in the game will be exactly $l_1, l_2, ..., l_m$ if all the rules are followed. If there are multiple solutions print any of them.

If there is no permutation which satisfies all described conditions print `-1`.

input

```
4 5
2 3 1 4 4
```

output

```
3 1 2 4
```

input

```
3 3
3 1 2
```

output

```
-1
```

Let's follow leadership in the first example:

- Child 2 starts.
- Leadership goes from 2 to $2 + a_2 = 3$.
- Leadership goes from 3 to $3 + a_3 = 5$. As it's greater than 4, it's going in a circle to 1.
- Leadership goes from 1 to $1 + a_1 = 4$.
- Leadership goes from 4 to $4 + a_4 = 8$. Thus in circle it still remains at 4.

C. Sofa Thief

1 second, 256 megabytes

Yet another round on DecoForces is coming! Grandpa Maks wanted to participate in it but someone has stolen his precious sofa! And how can one perform well with such a major loss?

Fortunately, the thief had left a note for Grandpa Maks. This note got Maks to the sofa storehouse. Still he had no idea which sofa belongs to him as they all looked the same!

The storehouse is represented as matrix $n \times m$. Every sofa takes two neighbouring by some side cells. No cell is covered by more than one sofa. There can be empty cells.

Sofa A is standing to the left of sofa B if there exist two such cells a and b that $\boldsymbol{x_a < x_b}$, a is covered by A and b is covered by B. Sofa A is standing to the top of sofa B if there exist two such cells a and b that $\boldsymbol{y_a < y_b}$, a is covered by A and b is covered by B. Right and bottom conditions are declared the same way.

Note that in all conditions $\boldsymbol{A \ne B}$. Also some sofa A can be both to the top of another sofa B and to the bottom of it. The same is for left and right conditions.

The note also stated that there are cnt_l sofas to the left of Grandpa Maks's sofa, cnt_r — to the right, cnt_t — to the top and cnt_b — to the bottom.

Grandpa Maks asks you to help him to identify his sofa. It is guaranteed that there is no more than one sofa of given conditions.

Output the number of Grandpa Maks's sofa. If there is no such sofa that all the conditions are met for it then output `-1`.

Input

The first line contains one integer number d ($1 \le d \le 10^5$) — the number of sofas in the storehouse.

The second line contains two integer numbers n, m ($1 \le n, m \le 10^5$) — the size of the storehouse.

Next d lines contains four integer numbers x_1, y_1, x_2, y_2 ($1 \le x_1, x_2 \le n$, $1 \le y_1, y_2 \le m$) — coordinates of the i-th sofa. It is guaranteed that cells (x_1, y_1) and (x_2, y_2) have common side, $(x_1, y_1) \ne (x_2, y_2)$ and no cell is covered by more than one sofa.

The last line contains four integer numbers cnt_l, cnt_r, cnt_t, cnt_b ($0 \le cnt_l, cnt_r, cnt_t, cnt_b \le d - 1$).

Output

Print the number of the sofa for which all the conditions are met. Sofas are numbered 1 through d as given in input. If there is no such sofa then print `-1`.

input

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

output

```
1
```

input

```
3
10 10
1 2 1 1
5 5 6 5
6 4 5 4
2 1 2 0
```

output

```
2
```

input

```
2
2 2
2 1 1 1
1 2 2 2
1 0 0 0
```

output

```
-1
```

Let's consider the second example.

- The first sofa has 0 to its left, 2 sofas to its right ($(1, 1)$ is to the left of both $(5, 5)$ and $(5, 4)$), 0 to its top and 2 to its bottom (both 2nd and 3rd sofas are below).
- The second sofa has $cnt_l = 2$, $cnt_r = 1$, $cnt_t = 2$ and $cnt_b = 0$.
- The third sofa has $cnt_l = 2$, $cnt_r = 1$, $cnt_t = 1$ and $cnt_b = 1$.

So the second one corresponds to the given conditions.

In the third example

- The first sofa has $cnt_l = 1$, $cnt_r = 1$, $cnt_t = 0$ and $cnt_b = 1$.
- The second sofa has $cnt_l = 1$, $cnt_r = 1$, $cnt_t = 1$ and $cnt_b = 0$.

And there is no sofa with the set $(1, 0, 0, 0)$ so the answer is `-1`.

D. Multicolored Cars

2 seconds, 256 megabytes

Alice and Bob got very bored during a long car trip so they decided to play a game. From the window they can see cars of different colors running past them. Cars are going one after another.

The game rules are like this. Firstly Alice chooses some color A, then Bob chooses some color B ($A \ne B$). After each car they update the number of cars of their chosen color that have run past them. Let's define this numbers after i-th car $cnt_A(i)$ and $cnt_B(i)$.

- If $cnt_A(i) > cnt_B(i)$ for every i then the winner is Alice.
- If $cnt_B(i) \ge cnt_A(i)$ for every i then the winner is Bob.
- Otherwise it's a draw.

Bob knows all the colors of cars that they will encounter and order of their appearance. Alice have already chosen her color A and Bob now wants to choose such color B that he will win the game (draw is not a win). Help him find this color.

If there are multiple solutions, print any of them. If there is no such color then print `-1`.

Input

The first line contains two integer numbers n and A ($1 \le n \le 10^5$, $1 \le A \le 10^6$) – number of cars and the color chosen by Alice.

The second line contains n integer numbers $c_1, c_2, ..., c_n$ $(1 \le c_i \le 10^6)$ — colors of the cars that Alice and Bob will encounter in the order of their appearance.

Output

Output such color B $(1 \le B \le 10^6)$ that if Bob chooses it then he will win the game. If there are multiple solutions, print any of them. If there is no such color then print `-1`.

It is guaranteed that if there exists any solution then there exists solution with $(1 \le B \le 10^6)$.

input
```
4 1
2 1 4 2
```
output
```
2
```

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

input
```
3 10
1 2 3
```
output
```
4
```

Let's consider availability of colors in the first example:

- $cnt_2(i) \ge cnt_1(i)$ for every i, and color 2 can be the answer.
- $cnt_4(2) < cnt_1(2)$, so color 4 isn't the winning one for Bob.
- All the other colors also have $cnt_j(2) < cnt_1(2)$, thus they are not available.

In the third example every color is acceptable except for 10.

E. Card Game Again

2 seconds, 256 megabytes

Vova again tries to play some computer card game.

The rules of deck creation in this game are simple. Vova is given an existing deck of n cards and a magic number k. The order of the cards in the deck is fixed. Each card has a number written on it; number a_i is written on the i-th card in the deck.

After receiving the deck and the magic number, Vova removes x (possibly $x = 0$) cards from the top of the deck, y (possibly $y = 0$) cards from the bottom of the deck, and the rest of the deck is his new deck (Vova has to leave at least one card in the deck after removing cards). So Vova's new deck actually contains cards $x + 1$, $x + 2$, ... $n - y - 1$, $n - y$ from the original deck.

Vova's new deck is considered *valid* iff the product of all numbers written on the cards in his new deck is divisible by k. So Vova received a deck (possibly not a *valid* one) and a number k, and now he wonders, how many ways are there to choose x and y so the deck he will get after removing x cards from the top and y cards from the bottom is *valid*?

Input

The first line contains two integers n and k $(1 \le n \le 100\,000,$ $1 \le k \le 10^9)$.

The second line contains n integers $a_1, a_2, ..., a_n$ $(1 \le a_i \le 10^9)$ — the numbers written on the cards.

Output

Print the number of ways to choose x and y so the resulting deck is *valid*.

input
```
3 4
6 2 8
```
output
```
4
```

input

```
3 6
9 1 14
```

output

```
1
```

In the first example the possible values of x and y are:

1. $x = 0, y = 0$;
2. $x = 1, y = 0$;
3. $x = 2, y = 0$;
4. $x = 0, y = 1$.

F. Level Generation

1 second, 256 megabytes

Ivan is developing his own computer game. Now he tries to create some levels for his game. But firstly for each level he needs to draw a graph representing the structure of the level.

Ivan decided that there should be exactly n_i vertices in the graph representing level i, and the edges have to be bidirectional. When constructing the graph, Ivan is interested in special edges called *bridges*. An edge between two vertices u and v is called a *bridge* if this edge belongs to every path between u and v (and these vertices will belong to different connected components if we delete this edge). For each level Ivan wants to construct a graph where at least half of the edges are *bridges*. He also wants to maximize the number of edges in each constructed graph.

So the task Ivan gave you is: given q numbers $n_1, n_2, ..., n_q$, for each i tell the maximum number of edges in a graph with n_i vertices, if at least half of the edges are *bridges*. **Note that the graphs cannot contain multiple edges or self-loops**.

Input

The first line of input file contains a positive integer q $(1 \le q \le 100\,000)$ — the number of graphs Ivan needs to construct.

Then q lines follow, i-th line contains one positive integer n_i $(1 \le n_i \le 2 \cdot 10^9)$ — the number of vertices in i-th graph.

Note that in hacks you have to use $q = 1$.

Output

Output q numbers, i-th of them must be equal to the maximum number of edges in i-th graph.

input

```
3
3
4
6
```

output

```
2
3
6
```

In the first example it is possible to construct these graphs:

1. $1 - 2, 1 - 3$;
2. $1 - 2, 1 - 3, 2 - 4$;
3. $1 - 2, 1 - 3, 2 - 3, 1 - 4, 2 - 5, 3 - 6$.

G. Four Melodies

5 seconds, 1024 megabytes

Author note: I think some of you might remember the problem "Two Melodies" from Eductational Codeforces Round 22. Now it's time to make it a bit more difficult!

Alice is a composer, and recently she had recorded two tracks that became very popular. Now she has got a lot of fans who are waiting for new tracks.

This time Alice wants to form four melodies for her tracks.

Alice has a sheet with n notes written on it. She wants to take four such non-empty non-intersecting subsequences that all of them form a *melody* and sum of their lengths is maximal.

Subsequence is a sequence that can be derived from another sequence by deleting some elements without changing the order of the remaining elements.

Subsequence forms a melody when each two adjacent notes either differ by `1` or are congruent modulo `7`.

You should write a program which will calculate maximum sum of lengths of such four non-empty non-intersecting subsequences that all of them form a melody.

Input

The first line contains one integer number n ($4 \le n \le 3000$).

The second line contains n integer numbers $a_1, a_2, ..., a_n$ ($1 \le a_i \le 10^5$) — notes written on a sheet.

Output

Print maximum sum of lengths of such four non-empty non-intersecting subsequences that all of them form a melody.

input

```
5
1 3 5 7 9
```

output

```
4
```

input

```
5
1 3 5 7 2
```

output

```
5
```

In the first example it is possible to compose 4 melodies by choosing any 4 notes (and each melody will consist of only one note).

In the second example it is possible to compose one melody with 2 notes — $\{1, 2\}$. Remaining notes are used in other three melodies (one note per each melody).