A. Maximal Binary Matrix

1 second, 256 megabytes

You are given matrix with n rows and n columns filled with zeroes. You should put k ones in it in such a way that the resulting matrix is symmetrical with respect to the main diagonal (the diagonal that goes from the top left to the bottom right corner) and is lexicographically maximal.

One matrix is lexicographically greater than the other if the first different number in the first different row from the top in the first matrix is greater than the corresponding number in the second one.

If there exists no such matrix then output `-1`.

Input

The first line consists of two numbers n and k ($1 \le n \le 100$, $0 \le k \le 10^6$).

Output

If the answer exists then output resulting matrix. Otherwise output `-1`.

input
2 1
output
1 0 0 0

input
3 2
output
1 0 0 0 1 0 0 0 0

input
2 5
output
-1

B. Distances to Zero

2 seconds, 256 megabytes

You are given the array of integer numbers $a_0, a_1, ..., a_{n-1}$. For each element find the distance to the nearest zero (to the element which equals to zero). There is at least one zero element in the given array.

Input

The first line contains integer n ($1 \le n \le 2 \cdot 10^5$) — length of the array a. The second line contains integer elements of the array separated by single spaces ($-10^9 \le a_i \le 10^9$).

Output

Print the sequence $d_0, d_1, ..., d_{n-1}$, where d_i is the difference of indices between i and nearest j such that $a_j = 0$. It is possible that $i = j$.

input
9 2 1 0 3 0 0 3 2 4
output
2 1 0 1 0 0 1 2 3

input
5 0 1 2 3 4
output
0 1 2 3 4

input
7 5 6 0 1 -2 3 4

output
2 1 0 1 2 3 4

C. Maximal GCD

1 second, 256 megabytes

You are given positive integer number n. You should create such **strictly increasing** sequence of k positive numbers $a_1, a_2, ..., a_k$, that their sum is equal to n and greatest common divisor is maximal.

Greatest common divisor of sequence is maximum of such numbers that every element of sequence is divisible by them.

If there is no possible sequence then output `-1`.

Input
The first line consists of two numbers n and k ($1 \le n, k \le 10^{10}$).

Output
If the answer exists then output k numbers — resulting sequence. Otherwise output `-1`. If there are multiple answers, print any of them.

input
6 3
output
1 2 3

input
8 2
output
2 6

input
5 3
output
-1

D. Magazine Ad

1 second, 256 megabytes

The main city magazine offers its readers an opportunity to publish their ads. The format of the ad should be like this:

There are space-separated non-empty words of lowercase and uppercase Latin letters.

There are hyphen characters `'-'` in some words, their positions set word wrapping points. Word can include more than one hyphen.

It is guaranteed that there are no adjacent spaces and no adjacent hyphens. No hyphen is adjacent to space. There are no spaces and no hyphens before the first word and after the last word.

When the word is wrapped, the part of the word before hyphen and the hyphen itself stay on current line and the next part of the word is put on the next line. You can also put line break between two words, in that case the space stays on current line. Check notes for better understanding.

The ad can occupy no more that k lines and should have minimal width. The width of the ad is the maximal length of string (letters, spaces and hyphens are counted) in it.

You should write a program that will find minimal width of the ad.

Input
The first line contains number k ($1 \le k \le 10^5$).

The second line contains the text of the ad — non-empty space-separated words of lowercase and uppercase Latin letters and hyphens. Total length of the ad don't exceed 10^6 characters.

Output
Output minimal width of the ad.

input
4 garage for sa-le
output
7

input
4 Edu-ca-tion-al Ro-unds are so fun

output
10

Here all spaces are replaced with dots.

In the first example one of possible results after all word wraps looks like this:

```
garage.
for.
sa-
le
```

The second example:

```
Edu-ca-
tion-al.
Ro-unds.
are.so.fun
```

E. Roma and Poker

2 seconds, 256 megabytes

Each evening Roma plays online poker on his favourite website. The rules of poker on this website are a bit strange: there are always two players in a hand, there are no bets, and the winner takes 1 virtual bourle from the loser.

Last evening Roma started to play poker. He decided to spend no more than k virtual bourles — he will stop immediately if the number of his loses exceeds the number of his wins by k. Also Roma will leave the game if he wins enough money for the evening, i.e. if the number of wins exceeds the number of loses by k.

Next morning Roma found a piece of paper with a sequence on it representing his results. Roma doesn't remember the results exactly, and some characters in the sequence are written in a way such that it's impossible to recognize this character, so Roma can't recall whether he won k bourles or he lost.

The sequence written by Roma is a string s consisting of characters `W` (Roma won the corresponding hand), `L` (Roma lost), `D` (draw) and `?` (unknown result). Roma wants to restore any *valid* sequence by changing all `?` characters to `W`, `L` or `D`. The sequence is called *valid* if all these conditions are met:

- In the end the absolute difference between the number of wins and loses is equal to k;
- There is no hand such that the absolute difference before this hand was equal to k.

Help Roma to restore any such sequence.

Input

The first line contains two numbers n (the length of Roma's sequence) and k ($1 \le n, k \le 1000$).

The second line contains the sequence s consisting of characters `W`, `L`, `D` and `?`. There are exactly n characters in this sequence.

Output

If there is no *valid* sequence that can be obtained from s by replacing all `?` characters by `W`, `L` or `D`, print `NO`.

Otherwise print this sequence. If there are multiple answers, print any of them.

input
3 2 L??
output
LDL

input
3 1 W??
output
NO

input

```
20 5
?LLLLLWWWWW?????????
```

output

```
WLLLLLWWWWWWWWLWLWDW
```

F. Coprime Subsequences

2 seconds, 256 megabytes

Let's call a non-empty sequence of positive integers $a_1, a_2 \dots a_k$ *coprime* if the greatest common divisor of all elements of this sequence is equal to 1.

Given an array a consisting of n positive integers, find the number of its *coprime* subsequences. Since the answer may be very large, print it modulo $10^9 + 7$.

Note that two subsequences are considered different if chosen indices are different. For example, in the array $[1, 1]$ there are 3 different subsequences: $[1]$, $[1]$ and $[1, 1]$.

Input

The first line contains one integer number n ($1 \le n \le 100000$).

The second line contains n integer numbers $a_1, a_2 \dots a_n$ ($1 \le a_i \le 100000$).

Output

Print the number of *coprime* subsequences of a modulo $10^9 + 7$.

input

```
3
1 2 3
```

output

```
5
```

input

```
4
1 1 1 1
```

output

```
15
```

input

```
7
1 3 5 15 3 105 35
```

output

```
100
```

In the first example *coprime* subsequences are:

1. 1
2. $1, 2$
3. $1, 3$
4. $1, 2, 3$
5. $2, 3$

In the second example all subsequences are *coprime*.

G. Periodic RMQ Problem

4 seconds, 512 megabytes

You are given an array a consisting of positive integers and q queries to this array. There are two types of queries:

- $1\ l\ r\ x$ — for each index i such that $l \le i \le r$ set $a_i = x$.
- $2\ l\ r$ — find the minimum among such a_i that $l \le i \le r$.

We decided that this problem is too easy. So the array a is given in a compressed form: there is an array b consisting of n elements and a number k in the input, and before all queries a is equal to the concatenation of k arrays b (so the size of a is $n \cdot k$).

Input

The first line contains two integers n and k ($1 \le n \le 10^5$, $1 \le k \le 10^4$).

The second line contains n integers — elements of the array b ($1 \le b_i \le 10^9$).

The third line contains one integer q ($1 \le q \le 10^5$).

Then q lines follow, each representing a query. Each query is given either as $1\ l\ r\ x$ — set all elements in the segment from l till r (including borders) to x ($1 \le l \le r \le n \cdot k$, $1 \le x \le 10^9$) or as $2\ l\ r$ — find the minimum among all elements in the segment from l till r ($1 \le l \le r \le n \cdot k$).

Output

For each query of type 2 print the answer to this query — the minimum on the corresponding segment.

input

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

output

```
1
3
```

input

```
3 2
1 2 3
5
2 4 4
1 4 4 5
2 4 4
1 1 6 1
2 6 6
```

output

```
1
5
1
```