A. Worms

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

It is lunch time for Mole. His friend, Marmot, prepared him a nice game for lunch.

Marmot brought Mole n ordered piles of worms such that i-th pile contains a_i worms. He labeled all these worms with consecutive integers: worms in first pile are labeled with numbers 1 to a_1, worms in second pile are labeled with numbers $a_1 + 1$ to $a_1 + a_2$ and so on. See the example for a better understanding.

Mole can't eat all the worms (Marmot brought a lot) and, as we all know, Mole is blind, so Marmot tells him the labels of the best juicy worms. Marmot will only give Mole a worm if Mole says correctly in which pile this worm is contained.

Poor Mole asks for your help. For all juicy worms said by Marmot, tell Mole the correct answers.

Input

The first line contains a single integer n ($1 \le n \le 10^5$), the number of piles.

The second line contains n integers $a_1, a_2, ..., a_n$ ($1 \le a_i \le 10^3$, $a_1 + a_2 + ... + a_n \le 10^6$), where a_i is the number of worms in the i-th pile.

The third line contains single integer m ($1 \le m \le 10^5$), the number of juicy worms said by Marmot.

The fourth line contains m integers $q_1, q_2, ..., q_m$ ($1 \le q_i \le a_1 + a_2 + ... + a_n$), the labels of the juicy worms.

Output

Print m lines to the standard output. The i-th line should contain an integer, representing the number of the pile where the worm labeled with the number q_i is.

Examples

input

```
5
2 7 3 4 9
3
1 25 11
```

output

```
1
5
3
```

Note

For the sample input:

- The worms with labels from $[1, 2]$ are in the first pile.
- The worms with labels from $[3, 9]$ are in the second pile.
- The worms with labels from $[10, 12]$ are in the third pile.
- The worms with labels from $[13, 16]$ are in the fourth pile.
- The worms with labels from $[17, 25]$ are in the fifth pile.

B. DZY Loves Sequences

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

DZY has a sequence a, consisting of n integers.

We'll call a sequence $a_i, a_{i+1}, ..., a_j$ $(1 \le i \le j \le n)$ a subsegment of the sequence a. The value $(j - i + 1)$ denotes the length of the subsegment.

Your task is to find the longest subsegment of a, such that it is possible to change at most one number (change one number to any integer you want) from the subsegment to make the subsegment strictly increasing.

You only need to output the length of the subsegment you find.

Input

The first line contains integer n $(1 \le n \le 10^5)$. The next line contains n integers $a_1, a_2, ..., a_n$ $(1 \le a_i \le 10^9)$.

Output

In a single line print the answer to the problem — the maximum length of the required subsegment.

Examples

input

```
6
7 2 3 1 5 6
```

output

```
5
```

Note

You can choose subsegment a_2, a_3, a_4, a_5, a_6 and change its 3rd element (that is a_4) to 4.

C. Hot Days

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

The official capital and the cultural capital of Berland are connected by a single road running through n regions. Each region has a unique climate, so the i-th $(1 \le i \le n)$ region has a stable temperature of t_i degrees in summer.

This summer a group of m schoolchildren wants to get from the official capital to the cultural capital to visit museums and sights. The trip organizers transport the children between the cities in buses, but sometimes it is very hot. Specifically, if the bus is driving through the i-th region and has k schoolchildren, then the temperature inside the bus is $t_i + k$ degrees.

Of course, nobody likes it when the bus is hot. So, when the bus drives through the i-th region, if it has more than T_i degrees inside, each of the schoolchild in the bus demands compensation for the uncomfortable conditions. The compensation is as large as x_i rubles and it is charged in each region where the temperature in the bus exceeds the limit.

To save money, the organizers of the trip may arbitrarily add or remove extra buses in the beginning of the trip, and between regions (of course, they need at least one bus to pass any region). The organizers can also arbitrarily sort the children into buses, however, each of buses in the i-th region will cost the organizers $cost_i$ rubles. Please note that sorting children into buses takes no money.

Your task is to find the minimum number of rubles, which the organizers will have to spend to transport all schoolchildren.

Input

The first input line contains two integers n and m $(1 \le n \le 10^5; 1 \le m \le 10^6)$ — the number of regions on the way and the number of schoolchildren in the group, correspondingly. Next n lines contain four integers each: the i-th line contains t_i, T_i, x_i and $cost_i$ $(1 \le t_i, T_i, x_i, cost_i \le 10^6)$. The numbers in the lines are separated by single spaces.

Output

Print the only integer — the minimum number of roubles the organizers will have to spend to transport all schoolchildren.

Please, do not use the `%lld` specifier to read or write 64-bit integers in C++. It is preferred to use cin, cout streams or the `%I64d` specifier.

Examples

input

```
2 10
30 35 1 100
20 35 10 10
```

output

```
120
```

input

```
3 100
10 30 1000 1
```

```
5 10 1000 3
10 40 1000 100000
```

output

```
200065
```

Note

In the first sample the organizers will use only one bus to travel through the first region. However, the temperature in the bus will equal $30 + 10 = 40$ degrees and each of 10 schoolchildren will ask for compensation. Only one bus will transport the group through the second region too, but the temperature inside won't exceed the limit. Overall, the organizers will spend $100 + 10 + 10 = 120$ rubles.

D. Swaps in Permutation

time limit per test: 5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are given a permutation of the numbers $1, 2, ..., n$ and m pairs of positions (a_j, b_j).

At each step you can choose a pair from the given positions and swap the numbers in that positions. What is the lexicographically maximal permutation one can get?

Let p and q be two permutations of the numbers $1, 2, ..., n$. p is lexicographically smaller than the q if a number $1 \le i \le n$ exists, so $p_k = q_k$ for $1 \le k < i$ and $p_i < q_i$.

Input

The first line contains two integers n and m $(1 \le n, m \le 10^6)$ — the length of the permutation p and the number of pairs of positions.

The second line contains n distinct integers p_i $(1 \le p_i \le n)$ — the elements of the permutation p.

Each of the last m lines contains two integers (a_j, b_j) $(1 \le a_j, b_j \le n)$ — the pairs of positions to swap. Note that you are given a `positions`, not the values to swap.

Output

Print the only line with n distinct integers p'_i $(1 \le p'_i \le n)$ — the lexicographically maximal permutation one can get.

Example

input

```
9 6
1 2 3 4 5 6 7 8 9
1 4
4 7
2 5
5 8
3 6
6 9
```

output

```
7 8 9 4 5 6 1 2 3
```

E. Array and Operations

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

You have written on a piece of paper an array of n positive integers $a[1], a[2], ..., a[n]$ and m *good* pairs of integers $(i_1, j_1), (i_2, j_2), ..., (i_m, j_m)$. Each *good* pair (i_k, j_k) meets the following conditions: $i_k + j_k$ is an odd number and $1 \le i_k < j_k \le n$.

In one operation you can perform a sequence of actions:

- take one of the *good* pairs (i_k, j_k) and some integer v ($v > 1$), which divides both numbers $a[i_k]$ and $a[j_k]$;
- divide both numbers by v, i. e. perform the assignments: $a[i_k] = \frac{a[i_k]}{v}$ and $a[j_k] = \frac{a[j_k]}{v}$.

Determine the maximum number of operations you can sequentially perform on the given array. Note that one pair may be used several times in the described operations.

Input

The first line contains two space-separated integers n, m $(2 \le n \le 100, 1 \le m \le 100)$.

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

The following m lines contain the description of *good* pairs. The k-th line contains two space-separated integers i_k, j_k $(1 \le i_k < j_k \le n, i_k + j_k$ is an odd number).

It is guaranteed that all the *good* pairs are distinct.

Output

Output the answer for the problem.

Examples

input

```
3 2
8 3 8
1 2
2 3
```

output

```
0
```

input

```
3 2
8 12 8
1 2
2 3
```

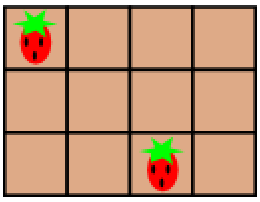
output

```
2
```

F. Cakeminator

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

You are given a rectangular cake, represented as an $r \times c$ grid. Each cell either has an evil strawberry, or is empty. For example, a 3×4 cake may look as follows:



The cakeminator is going to eat the cake! Each time he eats, he chooses a row or a column that does not contain any evil strawberries and contains at least one cake cell that has not been eaten before, and eats all the cake cells there. He may decide to eat any number of times.

Please output the maximum number of cake cells that the cakeminator can eat.

Input

The first line contains two integers r and c $(2 \le r, c \le 10)$, denoting the number of rows and the number of columns of the cake. The next r lines each contains c characters — the j-th character of the i-th line denotes the content of the cell at row i and column j, and is either one of these:

- '`.`' character denotes a cake cell with no evil strawberry;
- 'S' character denotes a cake cell with an evil strawberry.

Output

Output the maximum number of cake cells that the cakeminator can eat.

Examples

input

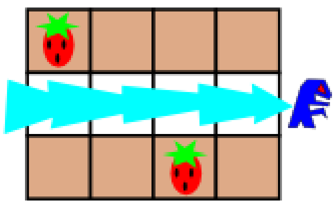
```
3 4
S...
....
..S.
```

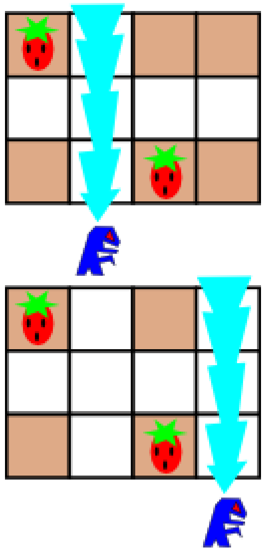
output

```
8
```

Note

For the first example, one possible way to eat the maximum number of cake cells is as follows (perform 3 eats).





G. Vasiliy's Multiset

time limit per test: 4 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Author has gone out of the stories about Vasiliy, so here is just a formal task description.

You are given q queries and a multiset A, initially containing only integer 0. There are three types of queries:

1. "+ x" — add integer x to multiset A.
2. "- x" — erase one occurrence of integer x from multiset A. It's guaranteed that at least one x is present in the multiset A before this query.
3. "? x" — you are given integer x and need to compute the value $\max_{y \in A}(x \oplus y)$, i.e. the maximum value of bitwise exclusive OR (also know as XOR) of integer x and some integer y from the multiset A.

Multiset is a set, where equal elements are allowed.

Input

The first line of the input contains a single integer q $(1 \le q \le 200\,000)$ — the number of queries Vasiliy has to perform.

Each of the following q lines of the input contains one of three characters '+', '-' or '?' and an integer x_i $(1 \le x_i \le 10^9)$. It's guaranteed that there is at least one query of the third type.

Note, that the integer 0 will always be present in the set A.

Output

For each query of the type '?' print one integer — the maximum value of bitwise exclusive OR (XOR) of integer x_i and some integer from the multiset A.

Example

input

```
10
+ 8
+ 9
+ 11
+ 6
+ 1
? 3
- 8
? 3
? 8
? 11
```

output

```
11
10
14
13
```

Note

After first five operations multiset A contains integers 0, 8, 9, 11, 6 and 1.

The answer for the sixth query is integer $11 = 3 \oplus 8$ — maximum among integers $3 \oplus 0 = 3$, $3 \oplus 9 = 10$, $3 \oplus 11 = 8$, $3 \oplus 6 = 5$ and $3 \oplus 1 = 2$.

H. Inverse Coloring

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are given a square board, consisting of n rows and n columns. Each tile in it should be colored either white or black.

Let's call some coloring *beautiful* if each pair of adjacent rows are either the same or different in every position. The same condition should be held for the columns as well.

Let's call some coloring *suitable* if it is *beautiful* and there is no **rectangle** of the single color, consisting of at least k tiles.

Your task is to count the number of *suitable* colorings of the board of the given size.

Since the answer can be very large, print it modulo 998244353.

Input

A single line contains two integers n and k ($1 \le n \le 500$, $1 \le k \le n^2$) — the number of rows and columns of the board and the maximum number of tiles inside the rectangle of the single color, respectively.

Output

Print a single integer — the number of *suitable* colorings of the board of the given size modulo 998244353.

Examples

input
```
1 1
```
output
```
0
```

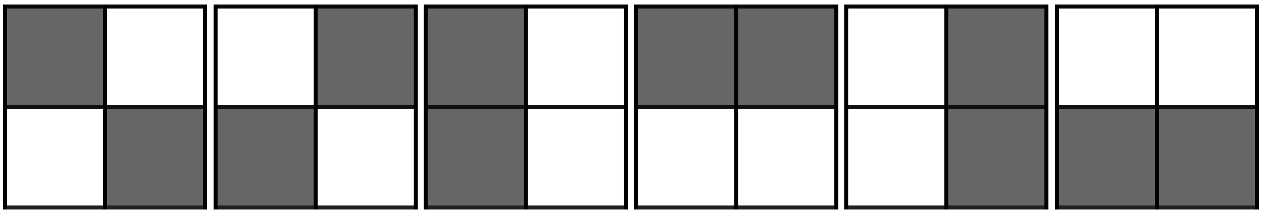
input
```
2 3
```
output
```
6
```

input
```
49 1808
```
output
```
359087121
```

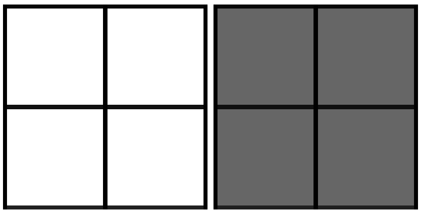
Note

Board of size 1×1 is either a single black tile or a single white tile. Both of them include a rectangle of a single color, consisting of 1 tile.

Here are the *beautiful* colorings of a board of size 2×2 that don't include rectangles of a single color, consisting of at least 3 tiles:



The rest of *beautiful* colorings of a board of size 2×2 are the following:

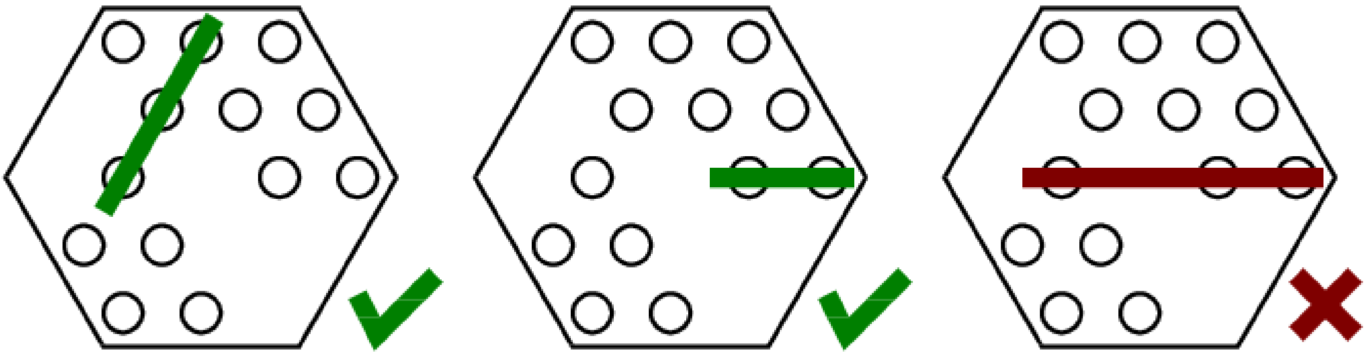


I. Sweets Game

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Karlsson has visited Lillebror again. They found a box of chocolates and a big whipped cream cake at Lillebror's place. Karlsson immediately suggested to divide the sweets fairly between Lillebror and himself. Specifically, to play together a game he has just invented with the chocolates. The winner will get the cake as a reward.

The box of chocolates has the form of a hexagon. It contains 19 cells for the chocolates, some of which contain a chocolate. The players move in turns. During one move it is allowed to eat one or several chocolates that lay in the neighboring cells on one line, parallel to one of the box's sides. The picture below shows the examples of allowed moves and of an unacceptable one. The player who cannot make a move loses.



Karlsson makes the first move as he is Lillebror's guest and not vice versa. The players play optimally. Determine who will get the cake.

Input

The input data contains 5 lines, containing 19 words consisting of one symbol. The word "`O`" means that the cell contains a chocolate and a "`.`" stands for an empty cell. It is guaranteed that the box contains at least one chocolate. See the examples for better understanding.

Output

If Karlsson gets the cake, print "`Karlsson`" (without the quotes), otherwise print "`Lillebror`" (yet again without the quotes).

Examples

input

```
  . . .
 . . O .
. . O O .
 . . . .
  . . .
```

output

```
Lillebror
```

input

```
  . . .
 . . . O
. . . O .
 O . O .
  . O .
```

output
Karlsson

J. Different is Good

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

A wise man told Kerem "Different is good" once, so Kerem wants all things in his life to be different.

Kerem recently got a string s consisting of lowercase English letters. Since Kerem likes it when things are different, he wants all *substrings* of his string s to be distinct. Substring is a string formed by some number of consecutive characters of the string. For example, string "`aba`" has substrings "" (empty substring), "`a`", "`b`", "`a`", "`ab`", "`ba`", "`aba`".

If string s has at least two equal substrings then Kerem will change characters at some positions to some other lowercase English letters. Changing characters is a very tiring job, so Kerem want to perform as few changes as possible.

Your task is to find the minimum number of changes needed to make all the substrings of the given string distinct, or determine that it is impossible.

Input

The first line of the input contains an integer n $(1 \le n \le 100\,000)$ — the length of the string s.

The second line contains the string s of length n consisting of only lowercase English letters.

Output

If it's impossible to change the string s such that all its substring are distinct print `-1`. Otherwise print the minimum required number of changes.

Examples

input

```
2
aa
```

output

```
1
```

input

```
4
koko
```

output

```
2
```

input

```
5
murat
```

output

```
0
```

Note

In the first sample one of the possible solutions is to change the first character to 'b'.

In the second sample, one may change the first character to 'a' and second character to 'b', so the string becomes "abko".

K. Sereja and Dima

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

Sereja and Dima play a game. The rules of the game are very simple. The players have n cards in a row. Each card contains a number, all numbers on the cards are distinct. The players take turns, Sereja moves first. During his turn a player can take one card: either the leftmost card in a row, or the rightmost one. The game ends when there is no more cards. The player who has the maximum sum of numbers on his cards by the end of the game, wins.

Sereja and Dima are being greedy. Each of them chooses the card with the larger number during his move.

Inna is a friend of Sereja and Dima. She knows which strategy the guys are using, so she wants to determine the final score, given the initial state of the game. Help her.

Input

The first line contains integer n $(1 \le n \le 1000)$ — the number of cards on the table. The second line contains space-separated numbers on the cards from left to right. The numbers on the cards are distinct integers from 1 to 1000.

Output

On a single line, print two integers. The first number is the number of Sereja's points at the end of the game, the second number is the number of Dima's points at the end of the game.

Examples

input

```
4
4 1 2 10
```

output

```
12 5
```

input

```
7
1 2 3 4 5 6 7
```

output

```
16 12
```

Note

In the first sample Sereja will take cards with numbers 10 and 2, so Sereja's sum is 12. Dima will take cards with numbers 4 and 1, so Dima's sum is 5.

L. Nearly Lucky Number

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Petya loves lucky numbers. We all know that lucky numbers are the positive integers whose decimal representations contain only the lucky digits **4** and **7**. For example, numbers **47**, **744**, **4** are lucky and **5**, **17**, **467** are not.

Unfortunately, not all numbers are lucky. Petya calls a number nearly lucky if the number of lucky digits in it is a lucky number. He wonders whether number n is a nearly lucky number.

Input

The only line contains an integer n $(1 \leq n \leq 10^{18})$.

Please do not use the %lld specificator to read or write 64-bit numbers in C++. It is preferred to use the cin, cout streams or the %I64d specificator.

Output

Print on the single line "YES" if n is a nearly lucky number. Otherwise, print "NO" (without the quotes).

Examples

input

```
40047
```

output

```
NO
```

input

```
7747774
```

output

```
YES
```

input

```
1000000000000000000
```

output

```
NO
```

Note

In the first sample there are 3 lucky digits (first one and last two), so the answer is "NO".

In the second sample there are 7 lucky digits, 7 is lucky number, so the answer is "YES".

In the third sample there are no lucky digits, so the answer is "NO".

M. Polo the Penguin and Matrix

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Little penguin Polo has an $n \times m$ matrix, consisting of integers. Let's index the matrix rows from 1 to n from top to bottom and let's index the columns from 1 to m from left to right. Let's represent the matrix element on the intersection of row i and column j as a_{ij}.

In one move the penguin can add or subtract number d from some matrix element. Find the minimum number of moves needed to make all matrix elements equal. If the described plan is impossible to carry out, say so.

Input

The first line contains three integers n, m and d $(1 \le n, m \le 100, 1 \le d \le 10^4)$ — the matrix sizes and the d parameter. Next n lines contain the matrix: the j-th integer in the i-th row is the matrix element a_{ij} $(1 \le a_{ij} \le 10^4)$.

Output

In a single line print a single integer — the minimum number of moves the penguin needs to make all matrix elements equal. If that is impossible, print "`-1`" (without the quotes).

Examples

input

```
2 2 2
2 4
6 8
```

output

```
4
```

input

```
1 2 7
6 7
```

output

```
-1
```