Problem A. Interesting numbers

Input: standard input
Output: standard output
Time limit per test: 1 second
Memory limit per test: 64 megabytes

There are natural numbers ending in **N**, such that the transfer of **N** to the beginning of the number leads to an increase in the number **N** times.

For example, number 102564, N=4, 410256=102564*4.

Find for a given number ***N*** the smallest natural number that satisfies this condition of the problem.

Input

One number is given $2 \le N \le 9$.

Output

Output one natural number.

Examples

Input	Output
4	102564

Remark. The required number may not fit into int64.

Problem B. Curious equation

Input: standard input
Output: standard output
Time limit per test: 1 second
Memory limit per test: 64 megabytes

Looking through a math book, the Lady found a curious equation of form ***A = S***. The equation is curious because **A** and **S** are not the same. The Lady realized that the left side of the equation must have additional operations between several pairs of of digits sequencies in **A**.

Write a program that inserts the least number of add operations on the left to make the equation correct. The numbers in the corrected equation can contain any number of leading zeros.

Input

The first line contains the equation in the form **A = S.**

A and **S** will be positive integers without leading zeros. They will be different.

A will contain no more than 1000 digits.

S will be less or equal to 5000.

Remark: the input data guarantees that the solution, although not necessarily unique, will always exist.

Output

Output the corrected equation. If there are several solutions, output any of them.

Examples

Input	Output
123=6	1+2+3=6
1015=25	10+15=25
2009=11	2+009=11

Problem C. Subpyramid

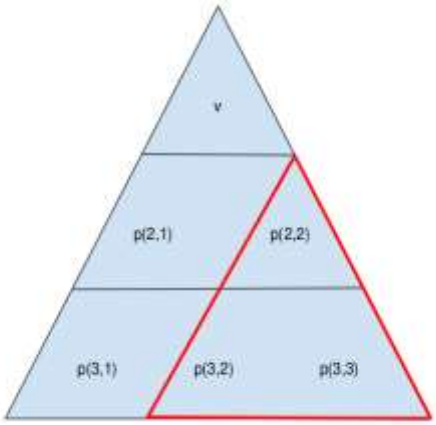
Input: standard input
Output: standard output
Time limit per test: 2 seconds
Memory limit per test: 128 megabytes

An arithmetic pyramid is a table of numbers **p (i, j)** defined for integers **1 ≤ j ≤ i ≤ n**. The **p (i, j)** value is defined as follows:

- p(1, 1) = v;
- p(i, 1) = (c · p(i − 1, 1)) mod m, is for 2 ≤ i ≤ n;
- p(i, j) = (a · p(i, j − 1) + b · p(i−1,j−1)) mod m, is for 2 ≤ i ≤ n and 2 ≤ j ≤ i.

For this arithmetic pyramid you need to answer queries about the maximum value in a given subpyramid. The subpyramid is determined by the coordinates of its vertex **(r, s)** and the side length **x.**

For example, in the figure below, the pyramid with r = 2, s = 2 and x = 2 is highlighted in red.



Input

The first line contains six integers: ***n, v, a, b, c*** and ***m (1 ≤ n ≤ 4000, 1 ≤ v, a, b, c ≤ 10^9, 2 ≤ m ≤ 10^9)*** - parameters of the pyramid.

The next line contains a single integer ***q (1 ≤ q ≤ 5 · 10^5)*** - number of requests.

This is followed by **q** lines, each of which contains three integers ***r, s, x (1 ≤ r ≤ n, 1 ≤ s ≤ r, 1 ≤ x ≤ n − r +1)*** - parameters of the subpyramid, which corresponds to the next request.

Output

For each inquiry, output the maximum value in the corresponding subpyramid.

Examples

Input	Output
3 1 1 1 1 3 2 1 1 1 1 1 2	1 2
5 3 1 1 1 5 4 1 1 1 3 3 2 1 1 4 2 1 1	3 4 4 3
34 72 111 13 17 912131 5 17 17 9 17 9 10 11 10 8 13 9 1 3 3 30	877499 896373 838846 429006 911914

Problem D. An old problem

Input: standard input
Output: standard output
Time limit per test: 4 second
Memory limit per test: 256 megabytes

Given an array of ***N*** numbers. You need to process 3 types of requests:

get(L, R, x) - say how many elements in the segment of the array ***[L..R]*** are at least ***x***.

set(L, R, x) - assign all elements of the array on the segment ***[L..R]*** ***x*** values

reverse(L, R) - invert a segment of an array ***[L..R]***. That is, the elements in places ***L*** and ***R*** will change places, as well as the elements in places ***L+1*** i ***R-1***, ***L+2*** i ***R-2*** etc.

Input

The first line contains a number ***N (1 ≤ N ≤ 100000)***. The second line contains an array of ***N*** numbers. Next, the number of queries is written ***M (1 ≤ M ≤ 30000)*** and ***M*** queries in the following lines. The format of the queries is as in the example. For all segments, ***1 ≤ L ≤ R ≤ N*** is correct. Array numbers and ***x*** numbers in queries are integers from ***0*** to ***1000000000***.

Output

For each request of the ***get*** type, it is necessary to output the answer.

Examples

Input	Output
5 1 2 3 4 5 6 get 1 5 3 set 2 4 2 get 1 5 3 reverse 1 2 get 2 5 2 get 1 1 2	3 1 3 1

Problem E. Black field

Input: standard input
Output: standard output
Time limit per test: 1 second
Memory limit per test: 64 megabytes

There is a field of size ***N*** by **N**, which consists of white **(.)** And black **(#)** cells. We can perform the operation ***OP (i, j)*** - take the ***i***-th line rotate 90 degrees clockwise and replace the ***j***-th column with it.

How many operations does it take to make a field completely black?

Input

The first line contains one integer ***N*** $(2 \le N \le 500)$.

Each of the next ***N*** lines contains ***N*** characters.

Output

Output one integer – the answer to the problem. If the solution does not exist, output -1.

Examples

Input	Output
2 #. .#	3

Problem F. Excursions

Input: standard input
Output: standard output
Time limit per test: 3 second
Memory limit per test: 64 megabytes

In Potokolandia there is a travel company that deals with tours of various types. On its website, the company posted its schedule in the form of an array A of N integers so that on the i-th day the company makes excursion A_i.

Nazar and Maxim are well-known travel enthusiasts. They want to choose a continuous sequence of days to visit Potokolandia. Nazar decided to arrive on day l_1 and stay until day r_1. As Nazar is a very busy person, it is known that he will stay in Potokolandia for no more than k nights. Maxim decided to visit Potokolandia from day l_2 to day r_2 inclusively. They will attend the corresponding excursions every day.

The boys are interested how many common types of excursions they will visit. You need to process q variants of given values l_1, r_1, l_2, r_2.

Input

The first line contains three integers N, q, k ($1 \le N, q \le 200000, \ 1 \le k \le 500$).

The second line contains N integers $A_1, A_2, \ldots, A_N$ ($1 \le A_i \le 10^9$).

Each of the next q lines contains an inquiry. Each inquiry contains four integers l_1, r_1, l_2, r_2 ($1 \le l_1 \le r_1 \le N, 1 \le l_2 \le r_2 \le N, r_1 - l_1 \le k$), as described in the condition.

Output

Output on a separate line for each inquiry the number of joint tours.

Examples

Input	Output
5 3 50 1 2 1 3 2 2 4 1 5 1 1 1 4 4 4 1 1	3 1 0

Problem G. Events

Input: standard input
Output: standard output
Time limit per test: 1 second
Memory limit per test: 32 megabytes

We consider a set of numbers from *1* to *n*. We can perform 2 different types of modifications and one type of inquiries on them:

- Modification of type D. We add the interval (x_i, y_i) to our set.
- Modification of type U. We remove the interval (x_i, y_i) from our set.
- An inquiry of type *S*. We ask whether there is a path from x_i to y_i, which means that there is a sequence a_1, a_2, ..., a_k ($k > 0$) of the existing intervals such that $a_i \cap a_{i+1} \notin \varnothing$ (i.e. that is, the intersection of the intervals a_i and a_{i+1} is not empty) and $x \in a_1, y \in a_k$ (i.e., the first interval contains the point *x*, and the last contains the point *y*).

Input

The first line contains two space-separated integers *n*, *q* ($1 \le n, q \le 500000$), corresponding to the range of elements and the number of inquiries, respectively. The next *q* lines contain inquiries. They are marked with the appropriate symbols.

1. For an inquiry *D* $1 \le x_i, y_i \le n$.
2. For an inquiry *U* the interval (x_i, y_i) is already in the set of intervals. If the interval has been specified multiple times, you only need to delete it once.
3. For an inquiry *S* $1 \le x_i, y_i \le n$.

Output

For each S-type event, output in a separate line "YES" if such a path exists, and "NO" if not.

Examples

Input	Output
10 10 S 5 5 D 1 4 D 5 9 S 4 5 D 4 5 S 3 7 U 5 9 S 5 5 U 1 4 U 4 5	NO NO YES YES

Problem H. Rectangles

Input: standard input
Output: standard output
Time limit per test: 1 second
Memory limit per test: 64 megabytes

You are given an $n \times m$ rectangle consisting of 1×1 squares. Some squares were cut out. You need to find the smallest number of rectangles into which you can cut the shape that remains.

Input

The first line contains two natural numbers n and m $(1 \le n, m \le 10)$. Next are n lines. The $(i + 1)$-th contains m numbers; the j-th number is equal to 1, if the cell in the **i**-th row and in the **j**-th column was cut from the board, 0 – if left.

Output

Output a single number - the minimum number of rectangles into which the resulting shape can be cut.

Examples

Input	Output
2 2 0 1 1 0	2