Educational Codeforces Round 65 (Rated for Div. 2)

A. Telephone Number

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

A telephone number is a sequence of exactly 11 digits, where the first digit is 8. For example, the sequence 80011223388 is a telephone number, but the sequences 70011223388 and 80000011223388 are not.

You are given a string s of length n, consisting of digits.

In one operation you can delete any character from string s. For example, it is possible to obtain strings 112, 111 or 121 from string 1121.

You need to determine whether there is such a sequence of operations (possibly empty), after which the string s becomes a telephone number.

Input

The first line contains one integer t ($1 \le t \le 100$) — the number of test cases.

The first line of each test case contains one integer n ($1 \le n \le 100$) — the length of string s.

The second line of each test case contains the string s ($|s| = n$) consisting of digits.

Output

For each test print one line.

If there is a sequence of operations, after which s becomes a telephone number, print YES.

Otherwise, print NO.

Example

input
2 13 7818005553535 11 31415926535
output
YES NO

Note

In the first test case you need to delete the first and the third digits. Then the string 7818005553535 becomes 88005553535.

B. Lost Numbers

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

This is an interactive problem. Remember to flush your output while communicating with the testing program. You may use `fflush(stdout)` in C++, `system.out.flush()` in Java, `stdout.flush()` in Python or `flush(output)` in Pascal to flush the output. If you use some other programming language, consult its documentation. You may also refer to the guide on interactive problems: https://codeforces.com/blog/entry/45307.

The jury guessed some array a consisting of 6 integers. There are 6 special numbers — 4, 8, 15, 16, 23, 42 — and each of these numbers occurs in a exactly once (so, a is some permutation of these numbers).

You don't know anything about their order, but you are allowed to ask **up to 4 queries**. In each query, you may choose two indices i and j ($1 \le i, j \le 6$, i and j are not necessarily distinct), and you will get the value of $a_i \cdot a_j$ in return.

Can you guess the array a?

The array a is fixed beforehand in each test, the interaction program doesn't try to adapt to your queries.

Interaction

Before submitting the answer, you may ask up to 4 queries. To ask a query, print one line in the following format: ? i j, where i and j should be two integers such that $1 \le i, j \le 6$. *The line should be ended with a line break character*. After submitting a query, flush the output and read the answer to your query — one line containing one integer $a_i \cdot a_j$. If you submit an incorrect query (or ask more than 4 queries), the answer to it will be one string 0. After receiving such an answer, your program should terminate immediately — otherwise you may receive verdict "Runtime error", "Time limit exceeded" or some other verdict instead of "Wrong answer".

To give the answer, your program should print one line ! a_1 a_2 a_3 a_4 a_5 a_6 *with a line break in the end*. After that, it should flush the output and terminate gracefully.

Example

input
16 64 345 672
output
? 1 1 ? 2 2 ? 3 5 ? 4 6 ! 4 8 15 16 23 42

Note

If you want to submit a hack for this problem, your test should contain exactly six space-separated integers a_1, a_2, ..., a_6. Each of 6 special numbers should occur exactly once in the test. The test should be ended with a line break character.

C. News Distribution

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

In some social network, there are n users communicating with each other in m groups of friends. Let's analyze the process of distributing some news between users.

Initially, some user x receives the news from some source. Then he or she sends the news to his or her friends (two users are friends if there is at least one group such that both of them belong to this group). Friends continue sending the news to their friends, and so on. The process ends when there is no pair of friends such that one of them knows the news, and another one doesn't know.

For each user x you have to determine what is the number of users that will know the news if initially only user x starts distributing it.

Input

The first line contains two integers n and m ($1 \le n, m \le 5 \cdot 10^5$) — the number of users and the number of groups of friends, respectively.

Then m lines follow, each describing a group of friends. The i-th line begins with integer k_i ($0 \le k_i \le n$) — the number of users in the i-th group. Then k_i **distinct** integers follow, denoting the users belonging to the i-th group.

It is guaranteed that $\sum_{i=1}^{m} k_i \le 5 \cdot 10^5$.

Output

Print n integers. The i-th integer should be equal to the number of users that will know the news if user i starts distributing it.

Example

input
7 5 3 2 5 4 0 2 1 2 1 1 2 6 7
output
4 4 1 4 4 2 2

D. Bicolored RBS

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

A string is called *bracket sequence* if it does not contain any characters other than "(" and ")". A bracket sequence is called *regular* (shortly, RBS) if it is possible to obtain correct arithmetic expression by inserting characters "+" and "1" into this sequence. For example, "", "(())" and "()()" are RBS and ")(" and "(()" are not.

We can see that each opening bracket in RBS is paired with some closing bracket, and, using this fact, we can define **nesting depth** of the RBS as maximum number of bracket pairs, such that the 2-nd pair lies inside the 1-st one, the 3-rd one — inside the 2-nd one and so on. For example, nesting depth of "" is 0, "()()()" is 1 and "()((())()())" is 3.

Now, you are given RBS s of even length n. You should color each bracket of s into one of two colors: red or blue. Bracket sequence r, consisting only of red brackets, should be RBS, and bracket sequence, consisting only of blue brackets b, should be RBS. Any of them can be empty. You are not allowed to reorder characters in s, r or b. No brackets can be left uncolored.

Among all possible variants you should choose one that **minimizes maximum** of r's and b's nesting depth. If there are multiple solutions you can print any of them.

Input
The first line contains an even integer n ($2 \le n \le 2 \cdot 10^5$) — the length of RBS s.

The second line contains regular bracket sequence s ($|s| = n$, $s_i \in \{$"(", ")"$\}$).

Output
Print single string t of length n consisting of "0"-s and "1"-s. If t_i is equal to 0 then character s_i belongs to RBS r, otherwise s_i belongs to b.

Examples

input
2 ()
output
11

input
4 (())
output
0101

input
10 ((()())())
output
0110001111

Note
In the first example one of optimal solutions is $s =$ "()". r is empty and $b =$ "()". The answer is $\max(0, 1) = 1$.

In the second example it's optimal to make $s =$ "(())". $r = b =$ "()" and the answer is 1.

In the third example we can make $s =$ "((()())(()))". $r =$ "()()" and $b =$ "(()())" and the answer is 2.

E. Range Deleting

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

You are given an array consisting of n integers $a_1, a_2, \ldots, a_n$ and an integer x. It is guaranteed that for every i, $1 \le a_i \le x$.

Let's denote a function $f(l, r)$ which erases all values such that $l \le a_i \le r$ from the array a and returns the resulting array. For example, if $a = [4, 1, 1, 4, 5, 2, 4, 3]$, then $f(2, 4) = [1, 1, 5]$.

Your task is to calculate the number of pairs (l, r) such that $1 \le l \le r \le x$ and $f(l, r)$ is sorted in non-descending order. Note that the empty array is also considered sorted.

Input
The first line contains two integers n and x ($1 \le n, x \le 10^6$) — the length of array a and the upper limit for its elements, respectively.

The second line contains n integers $a_1, a_2, \ldots a_n$ ($1 \le a_i \le x$).

Output
Print the number of pairs $1 \le l \le r \le x$ such that $f(l, r)$ is sorted in non-descending order.

Examples

input
3 3 2 3 1
output
4

input
7 4 1 3 1 2 2 4 3
output
6

Note

In the first test case correct pairs are $(1, 1)$, $(1, 2)$, $(1, 3)$ and $(2, 3)$.

In the second test case correct pairs are $(1, 3)$, $(1, 4)$, $(2, 3)$, $(2, 4)$, $(3, 3)$ and $(3, 4)$.

F. Scalar Queries

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

You are given an array $a_1, a_2, \ldots, a_n$. All a_i are pairwise distinct.

Let's define function $f(l, r)$ as follows:

- let's define array $b_1, b_2, \ldots, b_{r-l+1}$, where $b_i = a_{l-1+i}$;
- sort array b in increasing order;
- result of the function $f(l, r)$ is $\sum_{i=1}^{r-l+1} b_i \cdot i$.

Calculate $\left(\sum_{1 \le l \le r \le n} f(l, r)\right) \mod (10^9 + 7)$, i.e. total sum of f for all subsegments of a modulo $10^9 + 7$.

Input

The first line contains one integer n ($1 \le n \le 5 \cdot 10^5$) — the length of array a.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$, $a_i \ne a_j$ for $i \ne j$) — array a.

Output

Print one integer — the total sum of f for all subsegments of a modulo $10^9 + 7$

Examples

input
4 5 2 4 7
output
167

input
3 123456789 214365879 987654321
output
582491518

Note

Description of the first example:

- $f(1, 1) = 5 \cdot 1 = 5$;
- $f(1, 2) = 2 \cdot 1 + 5 \cdot 2 = 12$;
- $f(1, 3) = 2 \cdot 1 + 4 \cdot 2 + 5 \cdot 3 = 25$;
- $f(1, 4) = 2 \cdot 1 + 4 \cdot 2 + 5 \cdot 3 + 7 \cdot 4 = 53$;
- $f(2, 2) = 2 \cdot 1 = 2$;
- $f(2, 3) = 2 \cdot 1 + 4 \cdot 2 = 10$;
- $f(2, 4) = 2 \cdot 1 + 4 \cdot 2 + 7 \cdot 3 = 31$;

- $f(3, 3) = 4 \cdot 1 = 4$;
- $f(3, 4) = 4 \cdot 1 + 7 \cdot 2 = 18$;
- $f(4, 4) = 7 \cdot 1 = 7$;

G. Low Budget Inception

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

So we got bored and decided to take our own guess at how would "Inception" production go if the budget for the film had been terribly low.

The first scene we remembered was the one that features the whole city bending onto itself:

It feels like it will require high CGI expenses, doesn't it? Luckily, we came up with a similar-looking scene which was a tiny bit cheaper to make.

Firstly, forget about 3D, that's hard and expensive! The city is now represented as a number line (*infinite to make it easier, of course*).

Secondly, the city doesn't have to look natural at all. There are n buildings on the line. Each building is a square 1×1. **Buildings are numbered from 1 to n in ascending order of their positions.** Lower corners of building i are at integer points a_i and $a_i + 1$ of the number line. Also the distance between any two neighbouring buildings i and $i + 1$ doesn't exceed d (*really, this condition is here just to make the city look not that sparse*). Distance between some neighbouring buildings i and $i + 1$ is calculated from the lower right corner of building i to the lower left corner of building $i + 1$.

Finally, curvature of the bend is also really hard to simulate! Let the bend at some integer coordinate x be performed with the following algorithm. Take the ray from x to $+\infty$ and all the buildings which are on this ray and start turning the ray and the buildings counter-clockwise around point x. At some angle some building will touch either another building or a part of the line. You have to stop bending there (*implementing buildings crushing is also not worth its money*).

Let's call the angle between two rays in the final state the terminal angle α_x.

The only thing left is to decide what integer point x is the best to start bending around. Fortunately, we've already chosen m candidates to perform the bending.

So, can you please help us to calculate terminal angle α_x for each bend x from our list of candidates?

Input

The first line contains two integer numbers n and d ($1 \le n \le 2 \cdot 10^5$, $0 \le d \le 7000$) — the number of buildings and the maximum distance between any pair of neighbouring buildings, respectively.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($a_1 = 0$, $0 < a_{i+1} - a_i \le d + 1$) — coordinates of left corners of corresponding buildings in ascending order.

The third line contains single integer m ($1 \le m \le 2 \cdot 10^5$) — the number of candidates.

The fourth line contains m integers $x_1, x_2, \ldots, x_m$ ($0 \le x_i \le a_n + 1$, $x_i < x_{i+1}$) — the coordinates of bends you need to calculate terminal angles for in ascending order.

Output

Print m numbers. For each bend x_i print terminal angle α_{x_i} (in radians).

Your answer is considered correct if its absolute error does not exceed 10^{-9}.

Formally, let your answer be a, and the jury's answer be b. Your answer is accepted if and only if $|a - b| \leq 10^{-9}$.

Examples

input
3 5 0 5 7 9 0 1 2 3 4 5 6 7 8
output
1.570796326794897 1.570796326794897 0.785398163397448 0.927295218001612 0.785398163397448 1.570796326794897 1.570796326794897 1.570796326794897 1.570796326794897

input
2 7 0 4 3 1 3 4
output
1.570796326794897 0.927295218001612 1.570796326794897

input
5 0 0 1 2 3 4 6 0 1 2 3 4 5
output
1.570796326794897 3.141592653589793 3.141592653589793 3.141592653589793 3.141592653589793 1.570796326794897

Note

Here you can see the picture of the city for the first example and the bend at position 2 for it. The angle you need to measure is marked blue. You can see that it's equal to $\frac{\pi}{4}$.

You can see that no pair of neighbouring buildings have distance more than 4 between them. $d = 4$ would also suffice for that test.