Problem A. Image Scaling

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

You are given a $n \times m$ matrix consisting of . and `x`. All `x`s form a submatrix and you want to extract it and scale down as much as possible while keeping integer side length and unchanged aspect ratio. Output the final matrix.

Input

The first line contains two integers n, m.

The following n lines each contain a string of length m consisting of . and `x`.

Output

A matrix consisting of `x`.

Example

standard input	standard output
8 7 ....xxxx. ..xxxx. ..xxxx. ..xxxx. ..xxxx. ..xxxx.	xx xx xx

Note

$1 \leq n, m \leq 500$

Problem B. Break Sequence

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 seconds
Memory limit:	256 megabytes

Given a sequence a length n and a set S size m.

Count the number of sequences $0 = p_0 < p_1 < p_2 < \cdots < p_k < p_{k+1} = n$ such that:

- For any $0 \le i \le k, x \in S$ and $1 \le j \le n$, $\sum_{l=p_i+1}^{p_{i+1}} [a_l = j] \ne x$.

Input

Line 1: Two integers n, m.

Line 2: n integers, indicating sequence a.

Line 3: m integers, indicating set S.

Output

Line 1: An integer, the answer, modulo 998244353.

Examples

standard input	standard output
5 1 1 2 3 1 3 2	11
25 2 2 1 1 3 5 1 4 3 3 1 1 1 4 5 3 4 4 3 2 3 5 2 1 3 1 5 4	16411172

Note

- $1 \le n \le 2 \times 10^5$.
- $0 \le m \le 10$.
- $1 \le a_i \le n$.
- All integers in set S are between 1 and n.

Problem C. Change Matrix

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

There is an n×n matrix where the element at the i-th row and j-th column is $\gcd(i, j)(1 \le i, j \le n)$. There are q operations, each time multiplying all the numbers in a row or a column by a number, and outputting the sum of all numbers in the current matrix, modulo $10^9 + 7$, ensuring that the operations are random.

Input

The first line contains two positive integers n and q . The following q lines, each line is one of the following two operations respectively representing that all numbers in the x-th row or column need to be multiplied by y:

- R x y
- C x y

Output

In the following q lines, each line contains a single positive integer representing the answer.

Example

standard input	standard output
4 4 R 1 172460657 R 2 774542822 R 4 337889957 C 4 863617996	689842648 337099539 40219166 875100371

Note

$n, q \le 10^5, 1 \le x \le n, 0 \le y < 10^9 + 7$ It is guaranteed that all operations are chosen independently and randomly from all valid operations with equal probability and there are at most 10 test cases.

Problem D. Luner XOR

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Given the truth table of an n-variable Boolean function $f(x_1, x_2, \ldots, x_n) : F_2^n \to F_2$, determine its distance to all Boolean functions of degree at most 1, denoted as $g(x_1, x_2, \ldots, x_n) = a_0 \oplus a_1 x_1 \oplus a_2 x_2 \oplus \cdots \oplus a_n x_n$. The distance $d(f, g)$ is defined as $\sum_{x \in F_2^n}(f(x) \oplus g(x))$.

Input

The first line contains an integer n ($1 \le n \le 21$). The second line contains 2^n integers, indexed from 0. The i-th integer represents the value of $f(i \bmod 2, \lfloor \frac{i}{2} \rfloor \bmod 2, \lfloor \frac{i}{4} \rfloor \bmod 2, \ldots, \lfloor \frac{i}{2^{n-1}} \rfloor \bmod 2)$.

Output

To reduce the output size, you only need to output two integers: the sum of the squared distances to all Boolean functions of degree at most 1, modulo $10^9 + 7$, and the minimum distances to all Boolean functions of degree at most 1.

Examples

standard input	standard output
1 0 1	6 0
4 0 0 1 1 0 0 1 0 0 1 1 0 0 1 1 1	2176 4

Note

$1 \le n \le 21$.

Problem E. Easy Tree Problem

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Given a tree T with n vertices. The root of the tree is 1.

Each vertex has a value a_i and a limit b_i.

Given q queries of the following two types:

1. Change the value of vertex x to y.

2. Given v, perform the following operations:

- Let T' be T.
- Delete the edge between v and its father in T'.
- For each vertex i in subtree v in T, choose b_i sons and delete the edges between it and the rest of the sons in T'.
- Calculate the maximum sum of value of vertices in the component v.

Input

Line 1: An integer n.

Line 2: $n-1$ integers, the i-th one indicating the father of vertex $i+1$.

Line $3 \sim n+2$: 2 integers a_i and b_i on each line.

Line $n+3$: An integer q.

Line $n+4 \sim n+q+3$: Two or three integers. The first one *opt* indicating the type of query, followed by two integers x, y when $opt = 1$ or an integer v when $opt = 2$.

Output

For each query:

Line 1: An integer, the answer.

Examples

standard input	standard output
4 1 1 1 0 1 1 0 10 0 100 0 3 1 2 100 1 3 10 2 1	100
5 1 1 2 2 3 1 1 2 3 0 2 0 2 0 4 2 1 1 3 7 2 1 2 2	8 10 5
10 1 2 2 4 1 4 6 2 9 142704666 2 119707934 1 242911458 0 845889126 2 231420946 0 397143030 1 902907406 0 243233164 0 252401502 1 192973053 0 10 1 10 356281128 1 2 950761429 2 2 1 6 514594782 1 7 60062727 2 2 1 9 474318551 2 4 1 5 600976668 2 5	2930978907 2088134228 1137372799 600976668

Note

- $1 \leq n \leq 10^5$.

- $1 \le q \le 10^5$.
- $0 \le a_i, y \le 10^9$.
- b_i is no more than the number of sons of vertex i, and is positive if vertex i is not a leaf. $0 \le b_i$.
- $1 \le x, v \le n$.

Problem F. Minesweeper

Input file:	`standard input`
Output file:	`standard output`
Time limit:	6 seconds
Memory limit:	256 megabytes

You are given a rooted tree. Magically, for each vertex u and for each vertex v in the subtree of u, $u \le v < u + sz_u$ holds, where sz_u denotes the size of the subtree of u. Each vertex has a value a_i. At the beginning, $a_i = 0$.

In order to maintain it, you need to perform m operations. There are two types of operations:

- 1 l r x $(1 \le l, r, x \le n, l \le r)$ For each i such that $l \le i \le r$, assign $a_i + x \to a_i$.
- 2 u v $(1 \le u, v \le n)$ Print the sum of all a_w where w is in the path from u to v.

It's guaranteed that there will be no more than 10^5 operation 2.

Input

The first line contains two integers n, m $(1 \le n, m \le 10^6)$.

Next line contains $n-1$ integers $f_2, f_3, \ldots, f_n$, denoting the father of each vertex. The next m lines, each line contains an operation.

All inputted l, r, x, u, v are encoded, so you need to decode them by invoking $f(z) = (z \oplus \text{lastans}) \bmod n{+}1$ on them, where lastans denotes **the xor sum of all previous answers**. If there's no query operation before, lastans $= 0$. If you cannot understand, see the samples.

It's guaranteed that all numbers appeared are non-negative integers below 2×10^{18}.

Output

For each query, print one integer in one line, denoting the answer.

Example

standard input	standard output
6 8 1 2 2 4 4 1 12 9 10 1 13 9 0 2 10 11 1 14 13 14 1 0 6 11 2 9 2 1 11 11 2 2 4 15	6 12 30

Note

You may need to use fast input/output method.

Problem G. Sticky Pistons

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Given an array $a_1, \ldots, a_{n+1}$ of positive integers and a positive integer k. Initially, it is given that $a_i = i$ for each i. We define an operation as follows:

- Select any number of indices $1 \le i_1 < \ldots < i_m \le n$. Let j_p be the maximum index satisfying $a_{i_p} - i_p = a_{j_p} - j_p$.
- Initially, none of $a_1, \ldots, a_{n+1}$ is marked.
- For each $1 \le p \le m$ satisfying $j_p - i_p \le k$, mark $a_{i_p+1}, a_{i_p+2}, \ldots, a_{j_p}$ as **push.**
- **For each $1 \le p \le m$ satisfying $j_p - i_p \le k$, if a_{i_p} hasn't been marked, mark a_{i_p} as move.**
- **For each a_i that has been marked move, if $a_{i+1} = a_i + 2$ and a_{i+1} hasn't been marked, mark a_{i+1} as pull.**
- **Plus 1 to all a_i that have been marked push, and minus 1 to all a_i that have been marked pull.**

Your task is to make $a_{n+1} = 2n + 1$ within no more than Q_1 operations, as well as to make $a_{n+1} = n + 1$ within no more than Q_2 additional operations, where $Q_1 = n + \lfloor (n-1)/k \rfloor$ and $Q_2 = 2n - 1$.

Input

The first line contains a single integer t ($1 \le t \le 120$) — the number of test cases.

The first line of each test case contains two integers n ($1 \le n \le 1000$) and k ($1 \le k \le n$).

The sum of n over all test cases does not exceed 15000.

Output

The output of each test case contains two constructions with no more than Q_1 or Q_2 operations respectively.

The first line of each construction contains a non-negative integer Q — the number of operations you have used.

Each of the following Q lines contains several integers $m, i_1, \ldots, i_m$ separated by spaces, describing one operation.

operation	a after operation
—	$\{1, 2, 3\}$
$\{1\}$	$\{1, 3, 4\}$
$\{2\}$	$\{1, 3, 5\}$
$\{2\}$	$\{1, 3, 4\}$
$\{1\}$	$\{1, 2, 4\}$
$\{2\}$	$\{1, 2, 3\}$

Example

standard input	standard output
3 1 1 2 2 3 2	1 1 1 1 1 1 2 1 1 1 2 3 1 2 1 1 1 2 6 1 3 1 2 1 1 1 3 1 2 1 3 6 1 3 1 2 1 3 1 1 1 2 1 3

Note

In the second example, the array a after each operation is shown below:

In the third example, the construction may not be optimal.

Note that the movement of a sticky piston in this problem may be different from that in Minecraft.

Problem H. Two Convex Polygons

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	512 megabytes

In a two-dimensional plane, given two convex polygons (there may be three collinear points) A, B.

A is fixed. You can translate, rotate, flip B in the plane, but make sure that A and B have at least one point in common. Find the perimeter of the graph composed of points that B may cover.

Input

The first line is a positive integer T, which represents the number of test cases. For each test case:

- The first line contains a positive integer $n (3 \le n \le 10^6)$, representing the number of points in A convex polygon;
- For the following n lines, the i-th line contains two integers $x_i, y_i (-10^8 \le x_i, y_i \le 10^8)$ representing a vertex (x_i, y_i) of A convex polygon;
- The next line contains a positive integer $m (3 \le m \le 10^6)$, representing the number of points in B convex polygon;
- For the following m lines, the i-th line contains two integers $x'_i, y'_i (-10^8 \le x'_i, y'_i \le 10^8)$ representing a vertex (x'_i, y'_i) of B convex polygon.

Output

For each test case, output one line containing a real number representing the answer for this case. The answer will be accepted if and only if its absolute or relative error does not exceed 10^{-9}.

Example

standard input	standard output
1 3 0 0 3 0 3 4 4 0 0 3 0 3 4 0 4	43.41592653589793238469

Note

It is guaranteed that the sum of n over all test cases does not exceed 10^6 and the sum of m over all test cases does not exceed 10^6. The vertices of convex polygons are given in counterclockwise order and are different from each other. It is guaranteed that no convex polygon degenerates into a line segment.

Problem I. Interesting Numbers

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Little G is looking for a special type of number with the following characteristics: they have n digits. Splitting the number by half yields two perfect square numbers, possibly with leading zeros. It's guaranteed that $n \equiv 0 \bmod 2$.

Now Little G wants to know how many such numbers there are within the range $[L, R]$.

Input

Line 1: An integer n. Line 2: Two integers L and R.

Output

Line 1: An integer, the answer.

Examples

standard input	standard output
`2` `10 99`	`12`
`6` `100000 999999`	`704`

Note

$10, 11, 14, 19, 40, 41, 44, 49, 90, 91, 94, 99$ are the only numbers satisfying the conditions of the first sample.

$1 \leq n \leq 60, 10^{n-1} \leq L \leq R < 10^n$.

Problem J. String Problem

Input file:	standard input
Output file:	standard output
Time limit:	1 second
Memory limit:	512 megabytes

Seazycra doesn't know how to count, so he turned to studying string problems.

For a binary string p of length 2^k (indexed from 0, same below), if all binary strings S of length 10^9 with the first k positions and last k positions being 0 satisfy:

$$\sum_{i=0}^{10^9-k} p[f(S,i)] - \sum_{i=0}^{10^9-1} S_i = 0$$

Where S_i represents the number at the i-th position in S, $f(S,i) = \sum_{j=0}^{k-1} 2^{k-j-1} S_{i+j}$. Then p is called good.

Now Seazycra has an array q of length 2^k, he hopes to find a lexicographically smallest array p of length 2^k, such that $p \oplus q$ is good, where $\oplus$ denotes the XOR operation (e.g. $0011 \oplus 1010 = 1001$).

Input

The first line contains a integer k. The second line contains a binary string p of length 2^k. $2 \leq k \leq 20$.

Output

Output a single line containing a binary string q of length 2^k.

Examples

standard input	standard output
2 0000	0011
2 1111	1010

Problem K. Kill The Monsters

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

There are n monsters in the forest. The i-th one has a defense value of a_i.

You can perform the following attacks:

1. Decrease all monsters' defense values by 1. 2. Choose a monster and let its defense value a_i be $\lfloor \frac{a_i}{k} \rfloor$, where k is given.

Find the minimum number of operations required to make the defense value of each monster less than or equal to 0.

Input

Line 1: Two integers n, k.

Line 2: n integers, indicating sequence a.

Output

Line 1: An integer, the answer.

Examples

standard input	standard output
5 2 1 3 5 7 9	7
10 3 1 3 6 9 17 26 44 16 22 83	17

Note

- $1 \leq n \leq 10^5$.
- $1 \leq a_i, k \leq 10^9$.