Problem A. Wusu

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

Walker had a lot Wusu beer in Xinjiang. He discovered a fun fact about the beer: If one flips the bottle, the brand name "Wusu" becomes "nsnm" which is also a valid word in Chinese.

For any string s, we can rotate s by 180 degrees. (Equivalently, we first reverse the order of letters (reverse(s.begin(), s.end()); in C++) and then rotate each letter by 180 degrees.) The resulting string is called Wusu(s). For example, Wusu(bxq) is equal to bxq itself and Wusu(wma) is equal to ɐwm. If Wusu(s)=s, we say that s is a Wusu string. Given a string, find the length of its longest Wusu substring. (The empty string is a Wusu string and is a substring of any string.)

Note that in this problem, the input string consists of lower case English letters. We assume (b, q), (d, p), (w, m), (n, u) satisfy the condition that the first letter in each pair is equal to the second one in that pair after rotating 180 degrees and vice versa. o, s, x, z are equal to themselves respectively after rotating 180 degrees. Any letter other than the 12 letters mentioned above is no longer a letter after rotating 180 degrees.

Input

The only line contains a string s ($1 \le |s| \le 100000$).

Output

Output the length of the longest Wusu substring of s.

Examples

standard input	standard output
bxq	3
wma	2
wusu	1
nsnmwusu	8

Problem B. Multiplicative Function

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

In number theory, a multiplicative function is an arithmetic function $f(x)$ on positive integers with the property that $f(1) = 1$ and whenever a and b are coprime, $f(ab) = f(a)f(b)$.

Thus, if $n = \prod_{i=1}^{k} p_i^{e_i}$, where $p_1, p_2, \ldots, p_k$ are distinct primes and $e_1, \ldots, e_k$ are positive integers, $f(n) = \prod_{i=1}^{k} f(p_i^{e_i})$. f is determined uniquely by its values on prime powers.

You are given a function f by the values of $f(1), f(2), \ldots, f(n)$. We do not care about the value of f on integers larger than n. You need to support two operations:

1. Given a, b, modify $f(a) = b$. It is guaranteed that a is a prime power. Notice that if we change the value of $f(a)$, the value of $f(x)$ may change if a is a factor of x.

2. Compute $\sum_{i=1}^{m} f(i)$ modulo 2^{32}.

Input

The first line contains two integers n, q ($1 \le n \le 10^6, 1 \le q \le 10000$). n is defined in the statement and q is the number of operations. It is guaranteed that the number of sum queries (query 2) is at most 1000.

The second line contains n integers $f(1), \ldots, f(n)$. Each $f(i)$ is an integer between 0 and $2^{32} - 1$, inclusive. It is guaranteed that $f(ab) = f(a)f(b) \bmod 2^{32}$ for any coprime integers a and b in $[1, n]$ and that $f(1) = 1$.

Each of the next q lines contains an operation. "1 a b" represents the first type of operation ($2 \le a \le n$, a is a prime power, $0 \le b < 2^{32}$). "2 m" represents the second type of operation ($1 \le m \le n$).

Output

For each operation of the second type, output its answer in one line.

Example

standard input	standard output
10 7 1 2 3 4 5 6 7 8 9 10 2 10 1 2 10 2 10 1 5 100 2 10 1 4 0 2 10	55 127 1172 1168

Problem C. Cut the Cake

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

You are given 7 points on the plane. You need to find 3 lines to separate them if possible.

Formally, the points are separated by the lines if all the following conditions are satisfied:

1. No point lies on any line.
2. For any two different points a and b, there exists a line that intersects the line segment from a to b.

Input

The first line contains an integer T $(1 \leq 100)$ - the number of testcases.

For each testcase, there are 7 lines containing two integers x, y $(0 \leq x, y \leq 1000)$ each line denoting the coordinates of a point.

It is guaranteed that no three points are collinear.

Output

For each of the testcase, if it is impossible to find 3 lines to separate them, output 'No'.

Otherwise, output 'Yes' first. Then output three lines containing three integers a, b, c $(-10^9 \leq a, b \leq 10^9, -10^{18} \leq c \leq 10^{18})$ for the line $ax + by = c$.

Example

standard input	standard output
2 0 0 1 1 2 4 3 9 4 16 5 25 6 36 100 100 100 300 300 100 300 300 99 1000 1000 99 900 1000	No Yes 0 1 200 1 0 200 1 1 700

Problem D. Play with Intervals

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

You are given m intervals, each of them is represented as $[l_i, r_i]$ $(l_i \le r_i)$.

A set of intervals S is *excellent*, if for all $[l_1, r_1], [l_2, r_2] \in S$, at least one of the following conditions holds:

- $l_1 \le l_2 \le r_2 \le r_1$
- $l_2 \le l_1 \le r_1 \le r_2$
- $r_1 < l_2$
- $r_2 < l_1$

Note that in this problem, if $l_i = l_j$ and $r_i = r_j$ for some $i \ne j$, we still consider $[l_i, l_j]$ and $[r_i, r_j]$ as different intervals. In this case, the set S may contain both $[l_i, l_j]$ and $[r_i, r_j]$.

You want to figure out the maximum size among all *excellent* sets.

Input

There are multiple test cases. The first line of the input contains an integer T, indicating the number of test cases. For each test case:

The first line contains two integers n, m $(1 \le n, m \le 5000)$, the range and the number of intervals.

In the next m lines, there are two integers l_i, r_i $(1 \le l_i \le r_i \le n)$ in i-th line, indicating the i-th interval.

It is guaranteed that the sum of $max(n, m)$ over all test cases does not exceed 10000.

Output

For each test case, output the maximum size.

Example

standard input	standard output
2 3 3 1 2 1 2 2 3 5 4 3 5 1 4 2 2 4 5	2 3

Problem E. Building Subway Lines

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

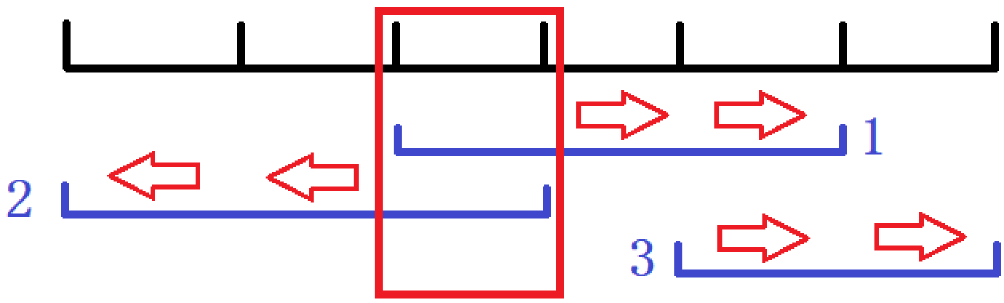
Metro system is a very important part in modern city, which can ease the traffic problem sufficiently. As transportation minister, you are planning to open m metro lines for your city.

The trunk road in your city can be divided into n sections, numbered from 1 to n. Subway line No.i runs from the L_i-th section to the R_i-th section. So you should build a tunnel for this subway line from section L_i to section R_i ($R_i - L_i + 1$ sections in total). Note that the subway lines are independent of each other, so you need to build their own tunnels separately.

You can do one of the following operations repeatedly:

- Spend a_j dollars to dig a very deep hole in section j, so that each subway line's tunnel in that section can be built. In other words, no matter how many subway lines contain section j, it only costs a_j dollars and the tunnels in section j of all subway lines are completed simultaneously.
- If for some subway line a, its tunnel in section $j-1$ has been built, you can spend d_j dollars to build a tunnel in section j for subway line b. Note that you can decide b freely. b can be equal to a and can also be not equal to a.
- If for some subway line a, its tunnel in section $j+1$ has been built, you can spend c_j dollars to build a tunnel in section j for subway line b. Note that you can decide b freely. b can be equal to a and can also be not equal to a.

You want to construct all the subway lines and minimize the total cost.



Input

The first line of the input contains two integers n, m ($1 \le n \le 10^6, 0 \le m \le \min(\frac{n(n+1)}{2}, 10^6)$), indicating the number of road sections and the number of subway lines.

The second line contains n integers, describing a_i ($1 \le a_i \le 10^9$).

The third line contains n integers, describing d_i ($1 \le d_i \le 10^6$).

The fourth line contains n integers, describing c_i ($1 \le c_i \le 10^6$).

In the next m lines, there are two integers L_i, R_i ($1 \le L_i \le R_i \le n$) in each line, describing the start section and the end section of subway line i.

Output

Output one integer, the minimum cost.

Example

standard input	standard output
6 3 5 5 1 3 5 7 5 5 5 2 2 2 2 2 2 5 5 5 3 5 1 3 5 6	13

Problem F. Cut the String

Input file:	`standard input`
Output file:	`standard output`
Time limit:	5 seconds
Memory limit:	256 megabytes

You are given a digit string $s_1 s_2 \dots s_n$. Each s_i is a decimal digit (one of $0, 1, 2, 3, 4, 5, 6, 7, 8, 9$). You can partition s into several substrings. (Partition s into only one piece is possible.) There are 2^{n-1} different partitions in total.

Each substring is interpreted as a decimal number. We define the weight of a partition as the product of all substrings in this partition.

Output the sum of weights of all partitions. Since the answer can be large, output the answer modulo 998244353.

Input

The first line contains a single integer n $(1 \leq n \leq 2 \times 10^5)$.

The second line contains the string $s = s_1 s_2 \dots s_n$. Each s_i is a decimal digit.

Output

Output one number in one line, the answer modulo 998244353.

Example

standard input	standard output
5 12345	41460

Problem G. Let's Fill the Binary String

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

Given a string $s_1 s_2 \cdots s_n$ of length n where $s_i \in \{‘0’, ‘1’, ‘?’\}$ and an integer k, please fill out all the '?'s with '0' or '1' such that the number of indices i satisfying $1 \le i < n$ and $s_i \ne s_{i+1}$ equals to k.

That's all? Nah! To heat things up (again) our dear friend Kotori asks you to find the answer with the smallest possible lexicographic order if it exists.

Recall that a string $a_1 a_2 \cdots a_n$ of length n is lexicographically smaller than another string $b_1 b_2 \cdots b_n$ of length n if there exists an integer k ($1 \le k \le n$) such that $a_i = b_i$ for all $1 \le i < k$ and $a_k < b_k$.

Input

There are multiple test cases. The first line of the input contains an integer T indicating the number of test cases. For each test case:

The first line contains two integers n and k ($1 \le n \le 10^5$, $0 \le k < n$) indicating the length of the string and the required number of indices satisfying the condition.

The second line contains a string $s_1 s_2, \cdots s_n$ ($s_i \in \{‘0’, ‘1’, ‘?’\}$).

It's guaranteed that the sum of n of all test cases will not exceed 10^6.

Output

For each test case output one line. If the answer exists output the lexicographically smallest one (you need to output the whole given string after filling out all the '?' and make this string the lexicographically smallest); Otherwise output "Impossible" (without quotes).

Example

standard input	standard output
5 9 6 1?010??01 9 5 1?010??01 9 6 100101101 9 5 100101101 9 3 ????????1	100100101 Impossible 100101101 Impossible 000000101

Problem H. Sum of Digits

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	256 megabytes

We define $s(x)$ as the sum of each digit of x in decimal. Given three integers L, R and d, your task is to calculate the number of integer pairs (a, b) satisfying the following two conditions:

1. $L \le a, b \le R$.
2. $s(a) + s(b) \equiv s(a + b) \pmod{d}$.

Input

The first line of input contains three integers d, L, R $(1 \le L \le R \le 10^{1000}, 2 \le d \le 9)$.

Output

Output one integers in one line — the number of integer pairs (a, b). Since the answer may be huge, you only have to ouput the answer modulo $10^9 + 7$.

Example

standard input	standard output
`3 10 100`	`8281`

Problem I. Walk

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

Define

$$f^1(x) = \begin{cases} x + y \ (x < k) \\ x - z \ (x \geq k) \end{cases}, f^s(x) = f^1(f^{s-1}(x))(s > 1).$$

Given x, y, z, k, s, please calculate $f^s(x)$.

Input

The first line contains one integer $test$ ($1 \leq test \leq 10^5$) – the number of test cases.

Each of the next $test$ lines contains five integers x, y, z, k, s ($1 \leq x, y, z, k, s \leq 10^9$).

Output

For each test case, we should output one integer $f^s(x)$.

Example

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

Problem J. Prime

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

Daddy Du has n integers $a_1, a_2, ..., a_n$.

For each operation, we can select one integer a_i $(1 \le i \le n)$ and modify it to $a_i + 1$ or $a_i - 1$ $(a_i > 1)$.

The sequence is *pang* if and only if for each i $(2 \le i \le n)$, $\gcd(a_{i-1}, a_i) = 1$.

Please calculate the minimum number of operations to make $a_1, a_2, ..., a_n$ *pang*.

Input

The first line contains one integer n $(1 \le n \le 10000)$.

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

The sequence is randomly generated.

Output

Output one integer – the minimum number of operations to make $a_1, a_2, ..., a_n$ *pang*.

Example

standard input	standard output
3 2 4 6	1

Problem K. Maximum Subset

Input file:	`standard input`
Output file:	`standard output`
Time limit:	6 seconds
Memory limit:	1024 megabytes

The map of Honno City can be described as a tree, i.e. a connected acyclic undirected graph.

A subset of the vertices is good if and only if the distances between each pair of vertices in the subset are not greater than d. Here the distance between two vertices means the number of edges on the simple path from one vertice to the other.

You are asked to find the largest good subset. Oh, there is an extra requirement! The mayor of Honno City likes one vertex so much that the subset must involve that vertex. However, the mayor's favorite vertex may change. So you need to find the largest good subset when the mayor's favorite vertex is $1, 2, 3, \cdots, n$.

Input

The first line contains two integers n, d ($1 \le n \le 3 \times 10^5, 0 \le d \le n$).

Each of the next $n - 1$ lines contains two integers x, y ($1 \le x, y \le n$) indicating there is an edge between vertex x and vertex y.

It is guaranteed the input graph is a connected acyclic undirected graph.

Output

Output n numbers in one line, the i-th number representing the size of the largest good set which involves vertex i.

Example

standard input	standard output
6 2 1 2 2 3 2 4 4 5 5 6	4 4 4 4 3 3