Problem A. Counting Pairs

Input file:	*standard input*
Output file:	*standard output*
Time limit:	4 seconds
Memory limit:	256 mebibytes

You are given an undirected graph G consisting of N vertices, numbered from 1 to N, and M edges.

Consider a pair of vertices (a, b), where $a < b$. Let the *incidence* of (a, b) be the total number of edges with at least one of their endpoints being a or b.

You have to answer Q queries. Each query is given as an integer k, and asks how many pairs of vertices (a, b) are there in G such that $a < b$ and the incidence of (a, b) is strictly greater than k.

Input

The first line of input contains two integers N and M, the number of vertices and the number of edges $(1 \leq N, M \leq 10^6)$.

Then M lines follow. The i-th of them contains two integers x_i and y_i, denoting the endpoints of the i-th edge $(1 \leq x_i, y_i \leq N)$. There may be self-loops or parallel edges.

The next line of input contains one integer Q, the number of queries $(1 \leq Q \leq 10^6)$.

Then Q lines follow. The i-th of them contains an integer k_i, denoting the i-th query $(1 \leq k_i \leq 10^6)$.

Output

For each query, print a single line with a single integer: the answer to the query.

Example

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

Problem B. Cactus

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

A *cactus* is a simple undirected connected graph in which every edge belongs to at most one simple cycle.

Now, there is a cactus accepting the following two operations:

1. Select a vertex with an odd degree in the graph, and remove all edges connected to it.
2. Make a copy of the current graph, and then draw additional edges between the corresponding vertices in the current graph and in the copy, forming a new graph. Formally speaking, suppose the current graph has n vertices in total, labeled from 1 to n. First, add n new vertices labeled from $n+1$ to $2n$. Then, for every edge (u, v) in the current graph, add an edge $(u+n, v+n)$. Lastly, add the edges $(1, n+1)$, $(2, n+2)$, $\ldots$, $(n, 2n)$. If the current graph has n vertices and m edges, the new graph has $2n$ vertices and $2m+n$ edges.

Because the second operation is costly, it can only be used at most once. The first operation can be used any number of times in any order.

Find a sequence of operations such that, after all operations in the sequence, the final graph has the least possible number of edges.

Input

The first line of input contains two integers n and m, the number of vertices and the number of edges in the initial graph ($1 \le n \le 3 \cdot 10^5$, $n-1 \le m \le \frac{3(n-1)}{2}$).

Each of the next m lines contains two integers u and v denoting the endpoints of an edge ($1 \le u, v \le n$). The graph is connected and contains no parallel edges and no self-loops.

Output

On the first line, print two integers m' and K, the number of edges left in the final graph and the total number of operations.

Then print K more lines. Each line represents an operation:

1. When using the first operation on vertex x, print "`1` x".
2. When using the second operation, just print "`2`".

If there are several optimal answers, print any one of them.

Examples

standard input	standard output
3 3 1 2 1 3 2 3	0 6 2 1 1 1 5 1 2 1 4 1 3
7 7 1 2 1 3 2 3 2 4 2 5 3 6 3 7	0 14 1 4 1 5 1 6 1 7 2 1 1 1 4 1 5 1 6 1 7 1 9 1 2 1 8 1 3

Problem C. Permute

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

Given a decimal integer s, you need to permute the digits of s to obtain a number divisible by 7, or determine that it is impossible.

Leading zeroes in decimal integers are allowed in this problem.

Input

The first line contains an integer T, the number of test cases ($1 \le T \le 10^5$). The descriptions of test cases follow.

A test case contains exactly one line with ten integers $c_0, c_1 \dots c_9$, where c_i is the number of digits i in s ($0 \le c_i \le 10^9$, $\sum c_i > 0$).

Output

For each test case, if it is possible, print the permuted number formatted according to the rules below. Otherwise, print -1.

Because the number s can be very large, you have to print the permuted number in segments, from left to right. First, print a line containing an integer k, the number of segments ($1 \le k \le 100$). Then print k more lines, the i-th of which will contain two integers r_i and x_i, indicating that the i-th segment of digits in the permuted number consists of r_i repetitions of the digit x_i ($r_i \ge 0$, $0 \le x_i \le 9$).

It can be shown that, if an answer exists, then there also exists an answer which can be represented under the above constraints. If there are several possible solutions, print any one of them.

Example

standard input	standard output
3 0 1 0 0 1 0 0 0 0 0 0 2 0 0 0 0 1 0 0 1 0 1000000000 0 0 0 0 0 0 0 0	2 1 1 1 4 3 2 1 1 6 1 9 -1

Problem D. Nimber Sequence

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

Given are Nimbers $a_1, a_2, \ldots, a_{K-1}$, $b_1, b_2, \ldots, b_5$, and $c_1, c_2, \ldots, c_5$.

Let $a_n = \left(\bigoplus_{i=1}^{5} a_{n-i} \otimes b_i \right) \oplus \left(\bigoplus_{i=1}^{5} a_{n-K+i} \otimes c_i \right)$ for all $n \geq K$.

Find the value of a_m.

Note

Nimbers are non-negative integers associated with Nim games. For the purposes of this problem, two operations are necessary:

- "$\oplus$" is the Nim sum: $a \oplus b = \operatorname{mex}(\{a' \oplus b \mid 0 \leq a' < a\} \cup \{a \oplus b' \mid 0 \leq b' < b\})$,
- "$\otimes$" is the Nim product: $a \otimes b = \operatorname{mex}(\{(a' \otimes b) \oplus (a \otimes b') \oplus (a' \otimes b') \mid 0 \leq a' < a,\ 0 \leq b' < b\})$.

Here, $\operatorname{mex}(S)$ represents the smallest non-negative integer $d \notin S$.

Input

The first line of input contains two positive integers K and m ($6 \leq K \leq 10^5$, $1 \leq m \leq 10^{18}$).

The second line contains $K - 1$ non-negative integers $a_1, a_2, \ldots, a_{K-1}$ ($0 \leq a_i < 2^{32}$).

The third line contains five non-negative integers $b_1, b_2, \ldots, b_5$ ($0 \leq b_i < 2^{32}$).

The fourth line contains five non-negative integers $c_1, c_2, \ldots, c_5$ ($0 \leq c_i < 2^{32}$).

Output

Output a single line with a single integer: the value of a_m.

Examples

standard input	standard output
6 1000000000000000000 1 2 3 4 5 1 0 0 0 0 0 0 0 0 0	5
6 10 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	9
11 123 849 674 223 677 243 657 979 583 643 845 979 282 313 567 433 122 443 132 554 132	32098

Problem E. Elephants

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3 seconds
Memory limit:	256 mebibytes

There are n elephants living in the grassland, numbered from 1 to n. Each elephant is either black or white. Unfortunately, you forgot all their individual colors.

You have observed these elephants for m days. On the i-th day, there was a group of k_i elephants $x_{i1}, x_{i2}, \ldots, x_{ik_i}$ hanging out. The thing you remember is that the absolute difference between the numbers of black and white elephants in each such group was at most 1.

You have also noticed that the elephants have a pattern of social activities. For any three elephants a, b, c, if a hangs out with b on day i and a hangs out with c on day j, then a hangs out with c on day i or a hangs out with b on day j, or both.

Can you find a possible coloring for all elephants?

Input

The first line of input contains two integers n and m, the number of elephants and the number of days ($1 \le n \le 10^6$, $0 \le m \le 10^6$).

Each of the following m lines contains an integer k_i followed by k_i distinct integers $x_{i1}, x_{i2}, \ldots, x_{ik_i}$ ($1 \le k_i \le n$, $\sum k_i \le 10^6$, $1 \le x_{ij} \le n$).

Output

Print a single line containing n binary digits separated by spaces. The i-th digit denotes the color of the i-th elephant: 0 for white or 1 for black.

If there are several possible solutions, print any one of them.

If there are no solutions, print a single integer -1 instead.

Example

standard input	standard output
5 4 3 1 4 5 2 1 5 2 2 3 1 3	1 0 1 1 0

Problem F. Interval Shuffle

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

Kanade has a sequence $A_{1\dots n}$ and m intervals $[L_i, R_i]$ of indices from 1 to n, bounds included. He does m operations in sequence, one for each interval. For the i-th operation, Kanade can choose and perform one of the following two actions:

1. Choose $x \in [L_i, R_i]$ and update $A_x := A_x + 1$.
2. Rearrange $A_{L_i \dots R_i}$ in any order Kanade wants.

Now Kanade wants to know the maximum value of A_k after these operations. Find the answer for each $k \in [1, n]$.

Input

The first line of input contains two integers n and m, the size of the sequence and the number of operations ($1 \le n, m \le 2 \cdot 10^5$). The second line contains n integers $A_{1\dots n}$, the initial sequence ($0 \le A_i \le 2 \cdot 10^5$).

Then follow m lines. The i-th of them contains two integers L_i and R_i describing the respective interval ($1 \le L_i \le R_i \le n$).

Output

Output n integers, the i-th of which is the maximum possible value of A_i after m operations.

Example

standard input	standard output
4 3 0 1 0 1 2 4 1 3 2 3	2 4 3 2

Problem H. Magic Box

Input file:	*standard input*
Output file:	*standard output*
Time limit:	5 seconds
Memory limit:	512 mebibytes

Rikka recently got a magic box. The box has n squares arranged in a row, and there is a lowercase English letter in each square. As a magician, Rikka can cast spells on the squares and give them magical power.

Firstly, she can choose a continuous sub-interval of squares and use a certain incantation to give these squares "the Power of Light". The incantation is just the concatenation of the letters on the chosen sub-interval of squares. For example, if Rikka chooses an interval of squares with letters '`a`', '`b`', and '`c`' written on them from left to right, then the incantation is "`abc`".

Secondly, she can choose a continuous sub-interval of squares (can be the same or different) and use a certain incantation to give these squares "the Power of Darkness". The incantation is also the concatenation of the letters on the chosen squares.

Lastly, the squares that have both powers simultaneously will be activated.

Rikka wants the two incantations to be exactly the same. She wants to know, for each number k from 0 to n, how many ways are there to use the same incantation twice and activate exactly k squares. Can you help her?

Input

The first line of input contains the string s consisting of lowercase English letters. The i-th letter of the string is the letter written in the i-th square from left to right. The length of the string is from 1 to $5 \cdot 10^5$ characters.

Output

Print a single line with $n+1$ integers, where the i-th integer is the number of ways to activate exactly $i-1$ squares. Here, n is the length of the input string.

Example

standard input	standard output
`aaaaa`	`13 9 6 4 2 1`

Problem I. Directed Acyclic Graph

Input file:	*standard input*
Output file:	*standard output*
Time limit:	5 seconds
Memory limit:	512 mebibytes

Recently, Rikka showed great interest in the data structures for directed acyclic graphs (DAGs). She dreams that extending classic tree-based algorithms like "weighted-chain decomposition" to their counterparts based on DAGs will be perfectly coooooool!

Now, she came up with a simple problem, and she would like to invite you to solve this problem with her.

You are given an n-node m-edge DAG G. Each node u has a non-negative integer value val_u. All values are set to 0 initially.

Rikka wants to perform q operations of three types described below:

1. Given u and x, set val_v to x for all v reachable from u;
2. Given u and x, set val_v to $\min\{val_v, x\}$ for all v reachable from u;
3. Given u, print its current value val_u.

Can you perform all these operations fast enough?

A node v is said to be *reachable* from u if there is a path starting in u and ending in v. A *path* is a node sequence $p_1, p_2, \ldots, p_k$ satisfying $(p_i, p_{i+1}) \in G$ for each $i = 1, 2, \ldots, k-1$.

Input

The first line of input contains three integers n, m, q ($1 \le n, m, q \le 10^5$).

Then m lines follow. Each of them contains two integers x and y, representing a directed edge (x, y) in the graph ($1 \le x, y \le n$). The input graph is guaranteed to be a DAG.

Then q lines follow. Each of them contains two or three integers in one of the following three formats:

- "`1` u x" indicating the first type of operation;
- "`2` u x" indicating the second type of operation;
- "`3` u" indicating the third type of operation.

All parameters in the operations above satisfy $1 \le u \le n$ and $0 \le x \le 10^9$.

Output

For each operation of the third type, print a single line containing an integer: the current value of val_u.

Example

standard input	standard output
4 4 7 1 2 1 3 3 4 2 4 1 1 5 1 2 1 3 3 3 4 2 1 3 3 2 3 3	5 1 1 3

Problem J. Paint

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

Zbox found an antique electronic painting board in a garbage dump. He discovered that it miraculously works! As the board is old-fashioned, Zbox can only paint with black and white colors on it.

Now Zbox tries to manipulate this painting board to draw some cool stuff.

This drawing board is completely white at the beginning, and supports the "Inverse Color" operation on a selected circle. In the i-th operation, Zbox specifies a circle with a radius of r_i centered at coordinates (x_i, y_i), and applies the "Inverse Color" tool. After the operation, the black part of this circle becomes white, and the white part becomes black.

For some reason, Zbox is concerned about the area of the black part of this painting board at each moment.

Zbox is nice! In order to avoid considering some corner cases, his operations guarantee some properties. See **data range** below for details.

Input

The first line contains an integer q indicating the total number of operations ($1 \le q \le 1000$).

Each of the next q lines contains three real numbers x, y, and r, where (x, y) is the center of the i-th circle selected for the "Inverse Color" operation, and r is its radius.

The **data range** for all data points guarantees that:

1. For any two circles specified for "Inverse Color", their centers do not coincide. Specifically, the distance between centers is greater than 10^{-4}.
2. For any two circles specified for "Inverse Color", their circumferences do not touch. Specifically, the distance between centers differs from the sum of the radii by more than 10^{-4}, and the distance between centers differs from the absolute difference of the radii by more than 10^{-4}.
3. The absolute value of every real number in the input data is at most 10^4.

Output

After each operation, print the total area of the black parts of the board. The answer will be considered correct if the absolute or relative error between the output and the standard answer is at most 10^{-6}.

Example

standard input	standard output
5 0 0 11 26 0 11 52 0 11 13 -11 11 39 -11 11	380.132711 760.265422 1140.398133 1331.290110 1522.182087