

Problem A. Triple-matching

Input file: `stdin`
Output file: `stdout`
Time limit: 0.5 s
Memory limit: 512 Mb

You are given a complete weighted tripartite graph. All the parts have the same size n . Tripartite means that there are no edges in the parts. Complete means that graph contains all possible edges between different parts (each edge only once).

You are to find triple-matching with minimal weight. Triple-matching is a set of n triples of indices (a_i, b_i, c_i) such that, all a_i are different, all b_i are different and all c_i are different. Weight of triple-matching is equal to $\sum_{i=1..n} (w[0, a_i, b_i] + w[1, b_i, c_i] + w[2, c_i, a_i])$, $w[i, a, b]$ — weight of edge from vertex with index a at part i to vertex with index b from part $(i + 1) \bmod 3$.

Input

The first line contains single integer $n \geq 1$.

The following n lines describe matrix $w[0]$.

The following n lines describe matrix $w[1]$.

The following n lines describe matrix $w[2]$.

It is guaranteed that all edge weights are random numbers between k and $2k$ for some k . All weights are integers between 0 and 10^5 inclusive.

Output

In the first line print weight of found triple-matching. On each of the following n lines print $a_i b_i c_i$ (vertices are indexed from 0 to $n - 1$). The triples may be printed in arbitrary order. If there are multiple triple-matchings with minimal weight, output any of them.

Scoring

Subtask 1 (25 points) $n \leq 5$.

Subtask 2 (25 points) $n \leq 9$.

Subtask 3 (25 points) $n \leq 11$.

Subtask 4 (25 points) $n \leq 13$.

Examples

stdin	stdout
3	611
82 100 98	2 0 2
95 71 75	1 1 1
55 97 61	0 2 0
87 99 72	
92 59 72	
55 80 64	
54 57 90	
74 60 77	
92 50 87	

Problem B. A Set of Subsets

Input file: `stdin`
Output file: `stdout`
Time limit: 0.5 s
Memory limit: 512 Mb

Given several random subsets of the set $A = \{1, 2, \dots, n\}$. You should choose the minimal number of subsets, whose union equals A . It is guaranteed, that the union of all the subsets equals A .

Input

The first line contains the integer $n \geq 1$ and the number of subsets $m \geq 1$. The following m lines give the subsets. Each subset is given by the number of elements $k_i \geq 1$ and k_i distinct integers from 1 to n . It is guaranteed, that the subsets were generated in the following way: fix $c \geq 5$, then m times at first choose a random size s from 1 to c , then uniformly from all C_n^s variants choose a set of size s .

Output

In the first line, output the minimal number of subsets x . In the next line, output the numbers of chosen subsets, x distinct integers from 1 to m . Numbers can be output in any order. If there are many optimal answers, output any of them.

Scoring

- Subtask 1 (25 points)** $n \leq 20, m \leq 50$.
Subtask 2 (25 points) $n \leq 30, m \leq 100$.
Subtask 3 (25 points) $n \leq 40, m \leq 100$.
Subtask 4 (25 points) $n \leq 50, m \leq 100$.

Examples

stdin	stdout
5 4	1
4 4 5 1 2	2
5 3 5 1 2 4	
1 2	
2 3 5	

Problem C. A Long Path

Input file: `stdin`
Output file: `stdout`
Time limit: 0.5 s
Memory limit: 512 Mb

Given a random undirected graph G with n vertices and m edges. Your task is to find a Hamiltonian path. It is guaranteed, that a Hamiltonian path exists in the graph.

Input

The first line contains the number of vertices $n \geq 2$ and the number of edges $m \geq 1$.

The next m lines contain pairs of integers from 1 to n , the edges of the graph.

There are no loops and multiple edges in the graph.

Since a nearly complete graph is uninteresting test, $m \leq 500$.

Output

In the first line, output n distinct numbers from 1 to n — vertices in a Hamiltonian path in a traversal order. You can start and finish at any vertices. If there are many Hamiltonian pathes, output any of them.

Scoring

- Subtask 1 (20 points)** $n \leq 26$.
- Subtask 2 (20 points)** $n \leq 35$.
- Subtask 3 (20 points)** $n \leq 50$.
- Subtask 3 (20 points)** $n \leq 70$.
- Subtask 4 (20 points)** $n \leq 100$.

Examples

stdin	stdout
5 8 3 1 2 5 5 4 3 4 1 4 3 5 3 2 1 2	1 4 3 5 2

Problem E. Roots

Input file: `stdin`
Output file: `stdout`
Time limit: 0.5 s
Memory limit: 512 Mb

Given an integer $n \geq 1$. You have to find such g , that for every a : $\gcd(a, n) = 1, 1 \leq a < n \implies \exists x: g^x = a \pmod n$. Recall that $\gcd(a, b)$ is the greatest common divisor of the numbers a and b .

Input

Attention, multi test case!

Each line contains an integer n ($2 \leq n \leq 10^{12}$).

We do not say, how many tests we have, but everything is beyond the bounds of decency.

Output

For each n , in a separate line output g ($1 \leq g < n$) or -1 , if such g does not exist.

Scoring

- Subtask 1 (15 points)** $n \leq 10^3$.
Subtask 2 (15 points) $n \leq 10^5$.
Subtask 3 (35 points) $n \leq 10^9$.
Subtask 4 (35 points) $n \leq 10^{12}$.

Examples

stdin	stdout
5	2
10	3
9	2
15	-1

Problem F. Multiplication of Polynomials

Input file: `stdin`
Output file: `stdout`
Time limit: `0.5 s`
Memory limit: `512 Mb`

Given polynomials $P(x)$ and $Q(x)$, find $P(x)Q(x)$. $Q(x) = x^k - 1$.

All the coefficients of $P(x)$ are random integers from 0 to $m - 1$.

All computations are performed modulo m , where m is a prime number from 2 to $10^9 + 7$.

Input

The first line contains integers n, k, m .

The next line contains n integers $p_0, p_1, \dots, p_{n-1}$, $P(x) = \sum p_i x^i$.

Output

Output $n + k$ integers from 0 to $m - 1$: $b_0, b_1, \dots, b_{n+k-1}$,

such that $P(x)Q(x) = \sum b_i x^i \pmod m$.

Scoring

Subtask 1 (50 points) $1 \leq n, k \leq 10^3$.

Subtask 2 (50 points) $1 \leq n, k \leq 10^5$.

Examples

<code>stdin</code>	<code>stdout</code>
4 1 7 1 2 3 4	6 6 6 6 4

Problem G. Division of Polynomials

Input file: `stdin`
Output file: `stdout`
Time limit: 0.5 s
Memory limit: 512 Mb

Given polynomials $P(x)$ and $Q(x)$, find such $A(x)$ and $R(x)$, that $P(x) = A(x)Q(x) + R(x)$ and $\deg R < \deg Q$. $Q(x) = x^k - 1$. All the coefficients of $P(x)$ are random integers from 0 to $m - 1$. All computations are performed modulo m , where m is a prime number from 2 to $10^9 + 7$.

Input

The first line contains integers n, k, m .

The next line contains n integers $p_0, p_1, \dots, p_{n-1}$, $P(x) = \sum p_i x^i$.

Output

In the first line, output $t = \max(n - k, 1)$ integers from 0 to $m - 1$: $a_0, a_1, \dots, a_{t-1}$. In the second line, output k integers from 0 to $m - 1$: $r_0, r_1, \dots, r_{k-1}$. The numbers should satisfy the following property: $P(x) = Q(x)(\sum a_i x^i) + \sum r_i x^i \pmod m$.

Scoring

Subtask 1 (50 points) $1 \leq n, k \leq 10^3$.

Subtask 2 (50 points) $1 \leq n, k \leq 10^5$.

Examples

stdin	stdout
3 1 7 1 5 1	6 1 0
3 1 7 2 4 1	5 1 0
3 1 7 0 4 1	5 1 5

Problem H. Edges are Being Added, Graph is Growing

Input file: `stdin`
Output file: `stdout`
Time limit: 0.5 s
Memory limit: 512 Mb

In an undirected graph, new edges are successively added. Initially the graph is empty. After each addition, you should say, if the current graph is bipartite.

Input

The first line contain integers n (the number of vertices) and m (the number adding an edge operations). The next m lines contain pairs of integers from 1 to n , descriptions of added edges.

Output

Output m zeros and ones in a row. The i -th symbol should be equal one, if the graph, consisting of the first i edges, is bipartite.

Scoring

- Subtask 1 (25 points)** $1 \leq n, m \leq 1\,000$.
Subtask 2 (50 points) $1 \leq n, m \leq 50\,000$.
Subtask 2 (25 points) $1 \leq n, m \leq 300\,000$.

Examples

stdin	stdout
3 3 1 2 2 3 3 1	110

Problem I. Sum of everything

Input file: `stdin`
Output file: `stdout`
Time limit: 0.5 s
Memory limit: 128 Mb

You are given random undirected graph. For each set of vertices A it is necessary to find $f(A)$ — number of independent sets of vertices B : $B \subseteq A$. Set of vertices B is independent, if there are no edge in graph such that B contains both edge vertices.

Input

The first line of input file contains number of vertices $n \geq 1$ and number of edges $m \geq 1$.

The next m lines contain pairs of integers between 1 and n — graph edges.

The graph does not contain loops (cycles of length 1) and multiple edges.

Output

For each set A there exists an integer $b(A)$, with binary representation corresponding to presence vertices in A .

Example: $n = 5, A = \{1, 2, 5\}, b(A) = 2^0 + 2^1 + 2^4 = 19$.

Output $\sum_A f(A) 2^{b(A)} \bmod (10^9 + 7)$

Scoring

Subtask 1 (25 points) $1 \leq n \leq 10$.

Subtask 2 (25 points) $1 \leq n \leq 16$.

Subtask 3 (30 points) $1 \leq n \leq 20$.

Subtask 4 (20 points) $1 \leq n \leq 23$.

Examples

stdin	stdout
3 1 1 2	1221

Explanation

The following set of vertices are independent: $\{\}, \{1\}, \{2\}, \{3\}, \{1, 3\}, \{2, 3\}$.

$A = \{\}$	$f(A) = 1$	$b(A) = 0$
$A = \{1\}$	$f(A) = 2$	$b(A) = 2^0 = 1$
$A = \{2\}$	$f(A) = 2$	$b(A) = 2^1 = 2$
$A = \{1, 2\}$	$f(A) = 3$	$b(A) = 2^0 + 2^1 = 3$
$A = \{3\}$	$f(A) = 2$	$b(A) = 2^2 = 4$
$A = \{1, 3\}$	$f(A) = 4$	$b(A) = 2^0 + 2^2 = 5$
$A = \{2, 3\}$	$f(A) = 4$	$b(A) = 2^1 + 2^2 = 6$
$A = \{1, 2, 3\}$	$f(A) = 6$	$b(A) = 2^0 + 2^1 + 2^2 = 7$

$$1 \cdot 2^0 + 2 \cdot 2^1 + 2 \cdot 2^2 + 3 \cdot 2^3 + 2 \cdot 2^4 + 4 \cdot 2^5 + 4 \cdot 2^6 + 6 \cdot 2^7 = 1221$$

Problem J. All Cliques

Input file: `stdin`
Output file: `stdout`
Time limit: 0.5 s
Memory limit: 512 Mb

An experienced DoTA player, Max, clicks a lot.

You are given a random undirected graph G with n vertices and m edges.

Subclique is a subset of vertices such that $A: \forall a, b \in A, a \neq b \quad \exists$ an edge (a, b) .

Subclique A is called maximal by inclusion, if $\nexists$ subclique $B \supset A: |B| > |A|$. Let $K(G)$ is amount of maximal by inclusion subcliques in G . Your task is to find $\min(K(G), 10^6)$.

Input

The first line of input file contains number of vertices $n \geq 1$ and number of edges $m \geq 1$.

The next m lines contain pairs of integers between 1 and n — graph edges.

The graph does not contain loops (cycles of length 1) and multiple edges.

Output

Print a single integer — $\min(K(G), 10^6)$.

Scoring

- Subtask 1 (15 points)** $n = 20$.
- Subtask 2 (15 points)** $n = 25$.
- Subtask 3 (20 points)** $n = 30$.
- Subtask 4 (15 points)** $n = 40$.
- Subtask 5 (20 points)** $n = 50$.
- Subtask 6 (15 points)** $n = 60$.

Examples

stdin	stdout
5 8 5 4 3 5 1 5 1 3 2 3 1 4 5 2 3 4	2

Problem K. Maximal clique

Input file: `stdin`
Output file: `stdout`
Time limit: 0.5 s
Memory limit: 512 Mb

An experienced DoTA player, Max, clicks a lot.

You are given a random undirected graph G with n vertices and m edges.
Subclique is a subset of vertices such that $A: \forall a, b \in A, a \neq b \quad \exists$ an edge (a, b) .
Your task is to find a subclique $A: |A|$ is maximal.

Input

The first line of input file contains number of vertices $n \geq 1$ and number of edges $m \geq 1$.

The next m lines contain pairs of integers between 1 and n — graph edges.

The graph does not contain loops (cycles of length 1) and multiple edges.

Output

In the first line print single number k — amount of vertices in the maximal subclique. In the next line print k integers between 1 and n — indices of vertices in the subclique in arbitrary order. If there are multiple optimal solutions, print any of them.

Scoring

- Subtask 1 (10 points)** $n \leq 20$.
- Subtask 2 (10 points)** $n \leq 25$.
- Subtask 3 (20 points)** $n \leq 40$.
- Subtask 4 (20 points)** $n \leq 60$.
- Subtask 5 (20 points)** $n \leq 70$.
- Subtask 6 (20 points)** $n \leq 80$.

Examples

stdin	stdout
5 8 5 4 3 5 1 5 1 3 2 3 1 4 5 2 3 4	4 1 3 4 5

Problem L. Learning to paint

Input file: `stdin`
Output file: `stdout`
Time limit: 0.5 s
Memory limit: 512 Mb

You are given a random undirected graph G with n vertices and m edges. Your task is to paint its vertices in minimal amount of colors such that each two adjanced vertices will be painted in different colors.

Input

The first line contains the number of vertices $n \geq 1$ and the number of edges $m \geq 1$.

The next m lines contain pairs of integers between 1 and n — graph edges.

The graph does not contain loops (cycles of length 1) and multiple edges.

Output

In the first line print single number k — minimal amount of used colors. In the next line print n integers between 1 and k — vertices colors. If there are multiple optimal solutions, print any of them.

Scoring

Subtask 1 (10 points) $n \leq 15$.

Subtask 2 (15 points) $n \leq 20$.

Subtask 3 (15 points) $n \leq 30$.

Subtask 4 (15 points) $n \leq 40$.

Subtask 5 (15 points) $n \leq 50$.

Subtask 6 (15 points) $n \leq 60$.

Subtask 7 (15 points) $n \leq 70$.

Examples

stdin	stdout
5 8 5 4 3 5 1 5 1 3 2 3 1 4 5 2 3 4	4 1 1 2 3 4

Problem M. The Easiest Way is to Paint in Three Colors

Input file: `stdin`
Output file: `stdout`
Time limit: 0.5 s
Memory limit: 512 Mb

You are given a random undirected graph G with n vertices and m edges. Your task is to paint its vertices in three colors such that each two adjacent vertices will be painted in different colors. It is guaranteed that such coloring exists.

Input

The first line contains the number of vertices $n \geq 1$ and the number of edges $m \geq 1$.

The next m lines contain pairs of integers between 1 and n — graph edges.

The graph does not contain loops (cycles of length 1) and multiple edges.

Output

Print n integers between 1 and 3 — vertices colors. If there are multiple such colorings, print any of them.

Scoring

- Subtask 1 (10 points)** $n \leq 20$.
- Subtask 2 (15 points)** $n \leq 30$.
- Subtask 3 (10 points)** $n \leq 45$.
- Subtask 4 (15 points)** $n \leq 64$.
- Subtask 5 (15 points)** $n \leq 90$.
- Subtask 6 (10 points)** $n \leq 120$.
- Subtask 7 (10 points)** $n \leq 170$.
- Subtask 8 (15 points)** $n \leq 250$.

Examples

stdin	stdout
5 5 1 2 2 3 3 1 4 5 1 5	1 2 3 2 3

Problem N. SAT USAT

Input file: `stdin`
Output file: `stdout`
Time limit: 0.5 s
Memory limit: 512 Mb

3-SAT is a well-known problem. Solve it. It is guaranteed that solution exists. **3-SAT** problem: there are m clauses, each consists of three conditions $x_i = 0$ and $x_i = 1$, it is necessary to find values of n boolean variables such that for each clause at least one of its free conditions will be true.

Input

The first line contains number of variables n and number of clauses m ($1 \leq m \leq \min(n^2, 1000)$).

Each of the following m lines contains numbers i, e_1, j, e_2, k, e_3 which correspond to conditions $x_i = e_1 \vee x_j = e_2 \vee x_k = e_3$.

$0 \leq e_1, e_2, e_3 \leq 1; 1 \leq i, j, k \leq n$. **All the clauses are randomly-created, however it is guaranteed that solution exists.**

Output

Output a single string of n zeroes and ones — values of the variables. If given **3-SAT** has multiple solutions, output any of them.

Scoring

- Subtask 1 (10 points)** $n \leq 20$.
- Subtask 2 (15 points)** $n \leq 30$.
- Subtask 3 (15 points)** $n \leq 40$.
- Subtask 4 (15 points)** $n \leq 50$.
- Subtask 5 (15 points)** $n \leq 70$.
- Subtask 6 (15 points)** $n \leq 100$.
- Subtask 7 (15 points)** $n \leq 160$.

Examples

stdin	stdout
2 3 1 0 1 0 1 0 2 0 2 1 1 1 1 1 2 1 1 1	01

Explanation

$(\underline{x_1 = 0} \vee \underline{x_1 = 0} \vee \underline{x_1 = 0}) \wedge (x_2 = 0 \vee \underline{x_2 = 1} \vee x_1 = 1) \wedge (x_1 = 1 \vee \underline{x_2 = 1} \vee x_1 = 1)$

In each clause true conditions are underlined.

Problem O. Almost bipartite beast

Input file: `stdin`
Output file: `stdout`
Time limit: 0.5 s
Memory limit: 512 Mb

You are given undirected, nonrandom graph. Vertices of the graph are divided into three sets A, B, X . $|X| = k \leq 10$. There are no edges between vertices in A , also there are no edges between vertices in B . Your task is to find a set of vertices Y with minimal size such that graph becomes bipartite after removing this set. Note that $|Y| \leq k$, because if set X will be removed the graph will be bipartite.

Input

The first line contains the number of vertices $n \geq 1$, the number of edges $m \geq 1$ and $k \geq 1$ — the size of the set X , $k \leq n$.

The next m lines contain pairs of integers between 1 and n — graph edges.

The graph does not contain loops (cycles of length 1) and multiple edges.

Vertices from sets A and B are numbered from 1 to $n - k$. Vertices of set X are numbered from $n - k + 1$ to n . Indices of vertices from A and B are shuffled. Intentionally, for your... convenience.

Output

In the first line print a single integer l — the size of the found set Y , on the second line print l different integers from 1 to n (elements of Y). If there are multiple optimal sets, output any of them.

Scoring

- Subtask 1 (20 points)** $n \leq 20, k \leq 10, m \leq 90$.
Subtask 2 (40 points) $n \leq 100, k \leq 10, m \leq 200$.
Subtask 3 (40 points) $n \leq 1000, k \leq 10, m \leq 2000$.

Examples

stdin	stdout
7 10 2 1 6 1 3 1 7 2 3 2 7 3 4 3 6 4 5 4 7 5 6	1 1
9 15 2 1 2 1 3 1 5 1 7 2 4 3 4 3 6 4 8 4 9 5 9 5 8 6 8 6 9 7 8 7 9	1 1