Problem A. Confusion

Input file:	`confusion.in`
Output file:	`confusion.out`
Time limit:	4 seconds
Memory limit:	256 mebibytes

Sasha lives in a hall, and he has a lot of things. Before leaving for the summer vacation, he faced the problem of packing things. Sasha has B boxes, T types of things, N things of these types and Q operations with them. Help him to handle all transactions. There are two types of operations:

1. Find out how many things of t type are in the k-box.
2. Take all things of type t from the top n items of the k-th box and put them on the top of the m-th box.

Input

The first line contains two integers: N ($1 \leqslant N \leqslant 10^5$), T ($1 \leqslant T \leqslant 10^5$).
The next line contains an integer B ($1 \leqslant B \leqslant 10^5$).
Each of the next B lines describes one box. It starts from an integer n ($0 \leqslant n \leqslant 10^5$) - number of things in this box. Next n integers describe types of things in this box from the bottom to the top. It's guaranteed that the total number of things in all boxes equals to N.
Next line contains an integer Q ($1 \leqslant Q \leqslant 10^5$).
Each of the next Q lines contains description of an operation. First of all z - type of operation. If $z = 1$ then this line contains two integers: t ($1 \leqslant t \leqslant T$), k ($1 \leqslant k \leqslant B$). If $z = 2$ then this line contains four integers: t ($1 \leqslant t \leqslant T$), k ($1 \leqslant k \leqslant B$), n ($0 \leqslant n \leqslant 10^5$), m ($1 \leqslant m \leqslant B$, $m \neq k$). It is guaranteed that the number of things in k-th box isn't less than n.

Output

You should output answer for all operations of the first type in separate lines.

Examples

confusion.in	confusion.out
5 2 2 3 2 2 1 2 1 2 5 1 2 1 1 2 2 2 2 1 2 2 1 2 1 1 2 2	2 1 1 2

Problem B. Good Old Table

Input file:	`cross.in`
Output file:	`cross.out`
Time limit:	2 seconds
Memory limit:	256 mebibytes

You are given a rectangular table of size $n \times m$ with ones and twos in each cell; it is guaranteed that $n \cdot m$ is even. You can change the table in steps. On each step, you choose one cell and invert all cells in its row and its column (ones become twos, twos become ones). Thus, on each step you invert exactly $n + m - 1$ elements. You should find the minimal number of steps required to transform all elements of the table to ones.

Input

The first line of input contains two integers n and m ($1 \leqslant n, m \leqslant 3000$, $n \cdot m$ is even). Next n lines describe the rows of the table. Each of these lines contains m integers separated by spaces. It is guaranteed that the elements of the table are ones and twos.

Output

On the first line of output, print the minimal number of steps required to transform all elements of the table to ones. If it is impossible, print -1.

Examples

cross.in	cross.out
2 2 1 2 2 1	2
3 4 1 2 1 2 1 1 2 2 2 1 1 2	3
1 4 2 1 1 1	-1

Note

In the second example, one of possible sequences of cells is $\{(1, 2), (2, 3), (3, 1)\}$.

Problem C. Radix

Input file:	`radix.in`
Output file:	`radix.out`
Time limit:	5 seconds
Memory limit:	256 mebibytes

In this problem, you are given a non-negative decimal integer (written in base 10), and your task is to print its binary representation (in base 2).

Input

The only one line of input contains the integer n ($0 \leqslant n \leqslant 10^{5 \cdot 10^5}$) without leading zeroes.

Output

On the first line of output, print the binary representation of n without leading zeroes.

Example

radix.in	radix.out
10	1010

Problem D. Random String Generator

Input file:	rsg.in
Output file:	rsg.out
Time limit:	2 seconds
Memory limit:	256 mebibytes

This problem is about generating strings. Let *random string generator* (RSG) be a program which generates a string consisting of characters A and B. String generation is a two-step process.

On the first step, the generator itself is generated. A parameter k, is chosen randomly from set $\{1, 2, \ldots, 10\}$ with equal probability. This parameter is the length of suffix which is sufficient to generate the next character.

After that, 2^k more parameters are chosen. These parameters are $p_s^{\mathtt{A}}$ for all strings s consisting of exactly k characters from set $\{\mathtt{A}, \mathtt{B}\}$. The $p_s^{\mathtt{A}}$ are chosen independently and uniformly on segment $[0, 1]$ (it means that probability of $p_s^{\mathtt{A}} < t$ equals t for every $t \in [0, 1]$). These parameters are the probabilities of letter A appearing after suffix s.

On the second step, we use the generator to generate an infinite string. The first k characters of the string are chosen independently and uniformly (each character is A with probability $\frac{1}{2}$).

Each next character depends only on the last k previous characters which form a suffix s of length k. This next character will be A with probability $p_s^{\mathtt{A}}$ and B with probability $p_s^{\mathtt{B}} = 1 - p_s^{\mathtt{A}}$.

You are given the first few characters of a string generated by the two-step process described above (note that the number of characters given could be less than k). You should output the probability that A was the next character of this string. It is guaranteed that the probability of generation of the given prefix is strictly greater than zero.

Input

The first line of input contains an integer T — the number of test cases ($1 \leqslant T \leqslant 10\,000$). Each of the next T lines contains a single test case — a nonempty string consisting only of characters A and B. All test cases were generated independently by the two-step process described above (in each test case, the generator and the infinite string are generated separately from other test cases). Sum of lengths of all T given strings does not exceed 10 000 characters.

Output

For each test case, print the required probability on a single line with absolute or relative error at most 10^{-6}.

Example

rsg.in	rsg.out
4 A BB AAA ABA	0.5 0.48333333333333334 0.5483870967741935 0.48333333333333334

Problem E. United States of Byteland

Input file:	scc.in
Output file:	scc.out
Time limit:	2 seconds
Memory limit:	256 mebibytes

In Byteland, there are N cities connected by two-way roads. King of Byteland decided to change the political system to federal constitutional republic with exactly K states (each state consists of several cities).

King doesn't want a revolution, so he asked his political scientists for help. They reported that the most simple way to reform the political system is to make each road a one-way road and divide cities into states according to the following simple rule: for each pair of cities A and B, they belong to the same state if and only if there are paths from A to B and from B to A. You are the best of the best programmers of the kingdom, so you should investigate if is it possible to reform without pain and horror.

Input

The first line of input contains three integers: N, the number of cities, M, the number of roads, and K, the required number of states. ($1 \leqslant N \leqslant 16$, $0 \leqslant M \leqslant 10^5$, $1 \leqslant K \leqslant N$). The next M lines contain two integers u_i and v_i — the numbers of cities connected by i-th road ($1 \leqslant u_i, v_i \leqslant N$).

Output

If the reform is impossible, print "NO". Other, on the first line, print "YES", and on the following M lines output the plan of the reform. Each of these M lines corresponds to one edge in the input. Edges should be output in the same order as in the input. The i-th of these lines should contain integers u_i and v_i in some order separated by a space. The order is important: if you output u_i v_i, i-th road is directed from u_i-th city to v_i-th city; if you output v_i u_i, it is directed from v_i-th city to u_i-th city instead.

Examples

scc.in	scc.out
5 6 3 1 2 2 3 3 1 1 4 2 5 4 5	YES 1 2 2 3 3 1 4 1 5 2 5 4
5 6 4 1 2 2 3 3 1 1 4 2 5 4 5	NO
16 0 16	YES

Problem F. Students' Life

Input file:	condoms.in
Output file:	condoms.out
Time limit:	2 seconds
Memory limit:	256 mebibytes

There is a party in a campus tonight! n male and n female students are going to take part in it. They are going to kiss each other, every boy is going to kiss every girl. But they care about safety, because they don't want to catch a cold. So they kiss only through a tissue to prevent exchange of microbes. A kiss is *safe* if every person kisses a side of tissue never kissed by anyone except him or her, and therefore, never take other person's microbes.

Students are rather tricky, so they can kiss through several tissues. In this case, sides of neighbouring tissues which are connected in such kiss take microbes from each other. For example, if male X kissed the right side of tissue A in one of previous kisses, he puts his microbes on this side, so if the right side of tissue A is connected now to the left side of tissue B, left side of tissue B take microbes of person X and becomes unsafe for kissing by anyone else.

Unfortunately they have only $\lfloor \frac{3n}{2} \rfloor$ tissues numerated from 1 to $\lfloor \frac{3n}{2} \rfloor$. You should output a sequence of $n \cdot n$ safe kisses between each pair of male and female students.

Input

The only line of input contains a positive integer n ($n \leqslant 1000$).

Output

Output a sequence of $n \cdot n$ lines with description of safe kisses. Each description is a line containing integers separated by single spaces. A description starts with a pair of integers a and b ($1 \leqslant a, b \leqslant n$) — indices of kissing male and female. Next goes the number k of tissues used in the kiss, followed by k pairs of integers c_i and d_i ($1 \leqslant c_i \leqslant \lfloor \frac{3n}{2} \rfloor$, $0 \leqslant d_i \leqslant 1$) — index of i-th tissue and side of i-th tissue which is closer to male. Tissues are listed in direction from male to female. All tissues within one kiss must be different.

You should output no more than $5 \cdot 10^7$ integers. It is guaranteed that such an answer exists given the constraints.

Example

condoms.in	condoms.out
2	1 1 2 1 0 2 0 1 2 1 1 0 2 1 1 2 0 2 2 2 2 0 1 0

Problem G. 3-substrings

Input file:	substr.in
Output file:	substr.out
Time limit:	2 seconds
Memory limit:	256 mebibytes

You are given a string S of length N. For each $K = 1, 2, \ldots, \lfloor \frac{N}{3} \rfloor$, you should find the number of different substrings of S of length exactly K such that each of them has at least three pairwise non-overlapping occurrences in S.

Input

The only line of input contains the string S ($3 \leqslant |S| \leqslant 100\,000$, S consists of lowercase English letters).

Output

You should output $\lfloor N/3 \rfloor$ numbers — answers for $K = 1$, $K = 2$, ..., $K = \lfloor N/3 \rfloor$.

Examples

substr.in	substr.out
abracadabra	1 0 0
abacabaabacabaabacaba	3 4 4 4 3 2 1
aaaa	1

Problem H. Tetrahedrons

Input file:	tetrahedron.in
Output file:	tetrahedron.out
Time limit:	3 seconds
Memory limit:	256 mebibytes

You are given a regular tetrahedron. Each of its edges is divided into n equal segments by $n - 1$ dividing points. Consider the set M of all dividing points. How many non-degenerate tetrahedrons with vertices from M exist?

Input

The single line of the input contains one integer n ($2 \leqslant n \leqslant 4000$).

Output

In the first line of output, print the answer to the problem.

Examples

tetrahedron.in	tetrahedron.out
2	12
37	65561472

Problem I. Ticket-punch

Input file:	`punch.in`
Output file:	`punch.out`
Time limit:	2 seconds
Memory limit:	256 mebibytes

Moscow Megapolice Departament of Public Transport decided to use ticket-punches in the airbuses and airmetro systems. Every passenger should insert rectangle ticket in ticket-punch and then receive it back with several square holes. A ticket-punch is a rectangle of $(2N + 1) \times (2M + 1)$ square cells. Let the rectangle rows be enumerated with integers from 1 to $2N + 1$ and columns with integers from 1 to $2M + 1$. Some cells with both even coordinates (at least one but, maybe, not all such cells) have square pins on them. These cells define the pattern of the ticket-punch. When a ticket is punched, each pin makes a square hole in the ticket. A ticket-punch can punch a ticket if every pin will make a hole, that is, every pin is strictly inside the ticket.

To prevent using one ticket more than once, each bus must have its own ticket-punch. Two ticket-punches are considered different if and only if their patterns cannot be matched using rotations, translations and/or reflections.

Given N and M, find the number of different ticket-punches of size $(2N + 1) \times (2M + 1)$ modulo $10^9 + 7$.

Input

The input contains several test cases. Each test case consists of one line containing two integers N and M ($1 \leqslant N, M \leqslant 1000$). The input file ends with the test case $N = M = 0$ which should not be processed. There are no more than 100 000 test cases in a single test (not including the terminating $N = M = 0$ case).

Output

For each test case, print the number of different ticket-punches of size $(2N+1) \times (2M+1)$ modulo $10^9 + 7$.

Example

punch.in	punch.out
2 2 2 3 0 0	5 19