Problem A. Asterisk

Input file:	`asterisk.in`
Output file:	`asterisk.out`
Time limit:	2 seconds
Memory limit:	64 megabytes

Recently Cuckooland mathematicians have invented a new binary operation «asterisk», which uses sequences as its arguments. Operation just appends the first sequence to the second. For example $(2,4)*(1,3) = (1,3,2,4)$. «Asterisk» operations in one expression are performed in order from the leftmost to the rightmost, but this order can be changed with brackets (operations in brackets are performed earlier). E. g. $(3) * ((1,5) * (2,7)) = (2,7,1,5,3)$. Notice that if a sequence element is represented by an expression, then this expression is calculated first and then all nested brackets in this sequence are removed. For example, $(1,(2*3),4) = (1,(3,2),4) = (1,3,2,4)$. The formal definition follows.

$$\begin{aligned}
\langle\text{expression}\rangle &\longrightarrow \langle\text{sequence}\rangle\ [*\langle\text{sequence}\rangle\ldots] \\
\langle\text{sequence}\rangle &\longrightarrow (\langle\text{sequence element}\rangle\ [,\langle\text{sequence element}\rangle\ldots]\,) \\
\langle\text{sequence element}\rangle &\longrightarrow \langle\text{number}\rangle \mid \langle\text{expression}\rangle \\
\langle\text{number}\rangle &\longrightarrow 1 \mid 2 \mid \ldots \mid N
\end{aligned}$$

Now cuckoolanders want to use this operation for generating permutations. More precisely, they want to obtain a given permutation from permutation $(1,2,\ldots,N)$ by adding brackets, commas and asterisks and evaluating the resulting expression.

Input

The first line contains an integer N ($1 \leqslant N \leqslant 10\,000$). The second line contains a permutation of numbers from 1 to N. These numbers are separated by spaces.

Output

Output a single line — correct expression, the result of which is the given permutation. Numbers from 1 to N should appear in ascending order. The length of the expression should not exceed 100 000 symbols. In case there is no such expression output «`IMPOSSIBLE`». Note that expression must not contain spaces and all sequences must be enclosed in brackets.

Examples

asterisk.in	asterisk.out
4 3 4 2 1	(1)*(2)*(3,4)
6 5 1 2 6 4 3	IMPOSSIBLE

Problem B. Anansi's Cobweb

Input file:	`cobweb.in`
Output file:	`cobweb.out`
Time limit:	2 seconds
Memory limit:	64 megabytes

Usatiy-Polosatiy XIII decided to destroy Anansi's home — his cobweb. The cobweb consists of N nodes, some of which are connected by threads. Let us say that two nodes belong to the same piece if it is possible to get from one node to the other by threads. Usatiy-Polosatiy has already decided which threads and in what order he would tear and now wants to know the number of pieces in cobweb after each of his actions.

Input

The first line contains integers N and M — the number of nodes and threads in the cobweb, respectively ($2 \leqslant N \leqslant 100\,000$, $1 \leqslant M \leqslant 100\,000$). Each of the next M lines contains two different integers — the 1-based indices of nodes connected by current thread. The threads are numbered from 1 to M in the order of description. Next line contains an integer Q which denotes the quantity of threads Usatiy-Polosatiy wants to tear ($1 \leqslant Q \leqslant M$). The last line contains numbers of these threads — different integers separated by spaces.

Output

Output Q integers — the number of pieces in Anansi's cobweb after each of Usatiy-Polosatiy's action. Separate numbers with single spaces.

Examples

cobweb.in	cobweb.out
4 4 1 2 2 3 1 3 3 4 3 2 4 3	1 2 3
3 1 1 2 1 1	3

Problem C. Droid Flies

Input file:	droids.in
Output file:	droids.out
Time limit:	5 seconds
Memory limit:	64 megabytes

Soon after the peace on Naboo planet had been restored, a gungan Dao-Dao decided to go hunting. However, he found a strange metal container with an emblem of the Trade Federation on it. Dao-Dao regarded it as trash and threw it into the forest. But the very next moment poor gungan was attacked by the whistling hive of droid flies — the modern weapon constructed by the Federation. He had to immediately alarm all gungans! But Dao-Dao decided to defeat the flies himself and create an indestructible power field around them. For the maximal stability the power field created by Dao-Dao must have a cubic shape, moreover, there must be at least one droid fly on each of the faces of the field. Help gungan to position this power field in space.

Input

The first line contains an integer N ($2 \leqslant N \leqslant 100$) — the number of droid flies. The following N lines contain the coordinates of the flies X_i, Y_i, Z_i — integers with absolute values not exceeding 100. No two flies are located at the same point.

Output

Output the coordinates of three normalized vectors collinear to the edges of the required cube. Output the answer as precisely as possible. You can assume that the required power field always exists.

Examples

droids.in	droids.out
6 0 1 1 1 0 1 1 1 0 1 1 2 1 2 1 2 1 1	1 0 0 0 1 0 0 0 1
2 1 0 0 -1 0 0	0.577350269 0.211324865 -0.788675135 0.577350269 0.577350269 0.577350269 0.577350269 -0.788675135 0.211324865

Problem D. Admission to Exam

Input file:	`exam.in`
Output file:	`exam.out`
Time limit:	3 seconds
Memory limit:	64 megabytes

At the end of the previous semester the students of the Department of Mathematics and Mechanics of the Yekaterinozavodsk State University had to take an exam in network technologies. N professors discussed the curriculum and decided that there would be exactly N^2 labs, the first professor would hold labs with numbers 1, $N+1$, $2N+1$, ..., N^2-N+1, the second one — labs with numbers 2, $N+2$, $2N+2$, ..., N^2-N+2, etc. N-th professor would hold labs with numbers N, $2N$, $3N$, ..., N^2. The professors remembered that during the last years lazy students didn't attend labs and as a result got bad marks at the exam. So they decided that a student would be admitted to the exam only if he would attend at least one lab of each professor.

N roommates didn't know the number of labs and professors in this semester. These students had different diligence: the first student attended all labs, the second one — only labs which numbers were a multiple of two, the third one — only labs which numbers were a multiple of three, etc... At the end of the semester it turned out that only K of these students were admitted to the exam. Find the minimal N which makes that possible.

Input

An integer K ($1 \leqslant K \leqslant 2 \cdot 10^9$).

Output

Output the minimal possible N which satisfies the problem statement. If there is no N for which exactly K students would be admitted to the exam, output 0.

Examples

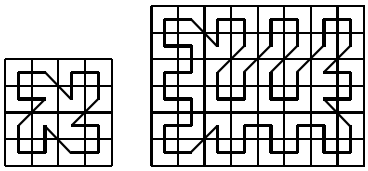
exam.in	exam.out
8	15
3	0

Problem E. Drunk King

Input file:	king.in
Output file:	king.out
Time limit:	2 seconds
Memory limit:	64 megabytes

Drunk king is a piece which moves as a usual chess king (i. e. to one of eight adjacent cells), but cannot make two consecutive moves in the same direction. Drunk king stands in an upper-left corner of an $N \times M$ chessboard and wants to visit each cell exactly once and return to initial position. His path must have no self-intersections.

Here are the examples of correct tours:



Help the king to find the required tour.

Input

The only input line contains 2 integers: N and M ($2 \leqslant N, M \leqslant 500$).

Output

In the first line output «Yes» or «No» depending on whether the required tour exists. If the tour exists, output it in the next $2N - 1$ lines with symbols «o» (ASCII code 111), «|» (code 124), «-» (code 45), «/» (code 47), «\» (code 92) and spaces. Each of these lines should contain exactly $2M - 1$ symbols. Use the format shown in the sample below. If there are many tours, you can output any of them.

Examples

king.in	king.out
6 8	Yes o-o o-o o-o o-o | \ | | | | | o-o o o o o o o | / / / / / o-o o o o o o-o | | | | | o-o o-o o-o o-o | \ o-o o o-o o-o o | /| | | | | o-o o-o o-o o-o
5 5	No

Problem F. Lunar Code 2

Input file:	`lunocode.in`
Output file:	`lunocode.out`
Time limit:	2 seconds
Memory limit:	64 megabytes

One may recall that a method of data encoding now known as lunar code was invented by lunar programmers during the defensive war against Martians. Even nowadays its slightly modified version is used by the Lunars in the data transmission. The data is represented in the form of matrix $M \times N$ containing ones and zeroes. The transferred matrix must satisfy the following check condition: exactly K of its rows and exactly L of its columns should contain zeroes only. If the received matrix does not satisfy this condition, then the data is considered to be corrupted during the transmission.

The Minister of Communications proposed to change the lunar code in his report for the President of the Lunar Federation. He claimed that the number of different messages that can be transmitted is not big enough. The president ordered the Ministry and the Lunar Academy of Sciences to research this question and to decide whether the code should be changed. It turned out that the minister was wrong because the number of binary matrices $M \times N$ satisfying the check condition is huge for big enough M and N. Can you calculate this number?

Input

The only input line contains 4 integers separated with space: M, N, K, L
($1 \leqslant M, N \leqslant 100\,000$; $0 \leqslant K \leqslant M$; $0 \leqslant L \leqslant N$).

Output

Output the number of matrices modulo $10^9 + 7$.

Examples

lunocode.in	lunocode.out
2 2 0 0	7
2 3 1 1	6

Problem G. Mortal Kombat

Input file:	mk.in
Output file:	mk.out
Time limit:	2 seconds
Memory limit:	64 megabytes

> Once every generation, there is a tournament known as Mortal Kombat, which was designed by the Elder Gods for the main purpose to save Earthrealm from the dark forces of Outworld. If the forces of Outworld win the tournament ten consecutive times, the Emperor will be able to invade and conquer Earthrealm. Thus far, Outworld has won nine straight victories, making the upcoming tournament the tenth, and possibly final one, for the Earthrealm.
>
> *From Wikipedia, the free encyclopedia*

There are N monsters and M best human fighters participating in the Mortal Kombat. According to the tournament rules, each monster should fight one of the humans (different monsters should fight different humans). If at least one monster wins, the Eathrealm will be conquered by the Emperor of the Outworld. However, the humans can choose the competitors and the order of battles.

The thunder god Raiden, protector of the Earthrealm, should choose the fighters in such a way that all Earth warriors will win their battles. For each monster and each Earth warrior it is known whether the Earth warrior can win the monster. First of all, the fighters for the first battle should be chosen.

For example, suppose that Liu Kang wants to fight Goro, but he is the only warrior able to defeat Shang Tsung, while Goro can be defeated by other warriors, such as Johnny Cage. So, even if Liu Kang will defeat Goro in the first battle, it will inevitably lead to the conquest of the Earth, because later Shang Tsung will defeat his opponent. This means that the pair Liu Kang vs. Goro should not be selected for the first fight.

Find out which pairs cannot be chosen by Raiden if he wants to save the freedom of humanity.

Input

The first line contains integers N and M ($1 \leqslant N \leqslant 300$, $N \leqslant M \leqslant 1500$). Next lines contain the binary matrix A with N rows and M columns. $A_{ij} = 1$ if and only if j-th Earth warrior can defeat i-th monster.

Output

Output matrix B with N rows and M columns. B_{ij} should be equal to one if the first battle cannot be held between i-th monster and j-th human, and zero otherwise.

Examples

mk.in	mk.out
4 4 1111 1000 1111 1111	1000 0111 1000 1000
4 5 10000 10000 10000 10000	11111 11111 11111 11111

Problem H. Monkey at the Keyboard

Input file:	monkey.in
Output file:	monkey.out
Time limit:	5 seconds
Memory limit:	64 megabytes

The «Entertaining math» showmen decided to ensure their audience in the well-known fact that a monkey randomly pressing keyboard buttons will sooner or later type the required word according to the probability theory.

The monkey taken to the shooting from the city zoo already can type — every second she types one of N first letters of English alphabet with equal probability. Fortunately, the word prepared by the showmen also contains some of these N letters only.

However, prior to giving a keyboard to the monkey, the showmen want to calculate the time it would take her to finish the job. More precisely, after how many seconds the given word will appear in the typed string for the first time?

Input

The first line contains an integer N — the number of letters the monkey can type ($1 \leqslant N \leqslant 26$). The second line contains a word proposed by the showmen. The word can contain only the first N lowercase Latin letters. Its length is positive and doesn't exceed 30 000.

Output

Output the expected time the monkey will need to type a word, rounded down to the nearest integer.

Examples

monkey.in	monkey.out
2 aa	6
2 ba	4

Problem I. Space Poker 3

Input file:	`poker.in`
Output file:	`poker.out`
Time limit:	5 seconds
Memory limit:	64 megabytes

Space poker. A legendary game, first version of which was introduced as far as in year 1284 of Alien era. Even nowadays its rules are known only to small group of professional players. Fortunately, the developers of the first program in the world playing space poker asked for your help.

There are N extraterrestial players in space poker. At the beginning of the round, each player gets M cards (we call them *hole cards*). Players don't know hole cards of their opponents. Then K *community* cards are consecutively dealt face-up. So, community cards are known to all players. Player's *hand* consists of his hole cards and all community cards — $M + K$ cards in total. There are no suits, cards differ only in their values. There are 13 different values: «`2`», «`3`», «`4`», ..., «`9`», «`T`», «`J`», «`Q`», «`K`» and «`A`». The card deck is infinite, and the probability of the event that the next card will have a given value is equal to 1/13. The combinations in space poker are represented in the form: $(v_1, \ldots, v_L)$, where L is the number of different values in the combination. The hand satisfies the combination $(v_1, \ldots, v_L)$ in case it contains v_1 cards of one value, v_2 cards of another value, ..., v_L cards of L-th value. For example, combination $(2, 2)$ is satisfied by hands «`2JA2A`» and «`22233`». Combination $(2, 3)$ is satisfied by hand «`KQKQKQ`» but is not satisfied by hand «`AAAAAA`». All combinations have different strength. The winner of the round is a player whose hand satisfies the combination of the maximal strength among all combinations in hands of all players. If there is more than one such player, the round ends in a draw.

Suppose you know the hole cards of the first player and partly dealt community cards. Calculate the probability the first player will be the only winner of this round.

Input

The first line contains integers N, M and K separated by spaces ($2 \leqslant N, M \leqslant 10$, $1 \leqslant K \leqslant 5$). The second line contains M symbols — hole cards of the first player. The third line contains at most K symbols — dealt community cards. The fourth line contains integer C — the number of combinations in space poker ($1 \leqslant C \leqslant 100$). The following C lines contain combinations in order of increasing strength. Each description has the form $L\ v_1\ v_2\ \ldots\ v_L$. L and v_i are positive integers, sum of all v_i doesn't exceed $M + K$.

Output

Output the probability of winning for the first player with absolute error not exceeding 10^{-5}.

Examples

poker.in	poker.out
2 5 2 23456 1 1 2	0.0883526857
2 5 2 23456 78 2 7 1 1 1 1 1 1 1 1 4	0.8407915043

Problem J. Scrooge's Tower

Input file:	`tower.in`
Output file:	`tower.out`
Time limit:	2 seconds
Memory limit:	64 megabytes

Scrooge McDuck owns a circular island of radius R centered at the origin. He wants to build a square tower there to keep all his wealth. The wealth is quite huge, so the square of the foundation should be inscribed in the circle of the island. After reading a lot of popular magazines, Mrs. Beakley claimed that according to feng shui, the entrance to the tower should be situated at the point (x_1, y_1), and the exit should be at the point (x_2, y_2). Of course, entrance and exit points should lie on the border of tower's foundation. Scrooge told you all these numbers and wants to know if it is possible to satisfy Mrs. Beakley's requirements.

Input

The first line contains an integer T — the number of test cases ($1 \leqslant T \leqslant 1\,000$). Each of the next T lines contains five integers separated by spaces: R, x_1, y_1, x_2, y_2 ($1 \leqslant R \leqslant 10\,000$). Points (x_1, y_1) and (x_2, y_2) are situated inside the circle.

Output

For each of the T cases output «`YES`» if the tower can be built, and «`NO`» otherwise.

Example

tower.in	tower.out
3 2 1 1 1 -1 2 1 1 0 0 1 1 0 -1 0	YES NO YES