Problem A. Different Words

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3 seconds
Memory limit:	512 mebibytes

Bytetec Bajtocki is a talented young computer specialist. The boy's parents realise how uncommonly skilled he is and often ask him for help with computer issues.

This time, Mr and Mrs Bajtocki decided to open an Internet banking account in Byteotian Bit Bank (BBB). To this end they both need to set their passwords to access the account. Bytetec's mom heard at the bank, that the safest passwords are those generated by a computer. Not knowing how to go about it, mom and dad turned to Bytetec for help.

BBB Internet banking service requires each password to be of minimum five characters. Upper and lowercase letters are permitted, as well as numerals and some punctuation marks — more specifically, all of ASCII characters described by codes from 48 to 122. In order his parents have no excessive trouble logging in, Bytetec decided to generate passwords consisting of exactly five characters for both of them.

Bytetec easily launched pseudorandom number generator and generated a certain number of possible passwords. Now he is considering which password should be assigned to his mom and which to his dad. Bytetec checked that for security reasons, the bank service requires that the passwords of individual account holders should be different. Bytetec decided to show off a bit in front of his parents and offered them a pair of truly different passwords, therefore passwords that differ with regards to each character position.

To make it easier for parents to remember their passwords, Bytetec would like mom's password to somehow resemble the name of the Bajtocki's family cat and dad's password — the name of their dog. The final choice of passwords will be done by Bytetec himself, but for this purpose he could use a list of all truly different passwords pairs from the list. Write a program that will generate such a list.

Input

The first line of input contains one integer n ($2 \le n \le 50\,000$) representing the number of passwords generated by Bytetec. Each of the subsequent n lines contain five ASCII characters with codes between 48 and 122. It is not to be assumed that these passwords are in any way random.

Output

The first line of output should contain a non-negative integer m representing the number of unordered pairs of truly different passwords. Each of the next m lines should contain two integers denoting the numbers of the two truly different passwords. Passwords are numbered from 1 to n as they appear on input.

If among passwords at the input there are more than 100 000 truly different pairs, only any 100 000 of such pairs should be presented (and then as m 100 000 should be presented).

Examples

standard input	standard output
3 aB;Va xBx@a zc:ng	2 1 3 2 3

Problem B. Electrical Circuit

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3 seconds
Memory limit:	512 mebibytes

Captain Byteasar currently explores planet *Vicugna*, which is highly unusual in every respect — besides many interesting facts (such as the presence of pink lam species), the planet has a variable magnetic field. In order to investigate magnetic phenomena Byteasar chose n nodes on the planet and connected them with m electrical wires. Those wires, thanks to magnetic field effect, carried a current of a certain value, possibly different for various cables. Current flows in accordance with the laws of physics — it flows in one direction in each wire, the sum of currents flowing into every node is equal to the sum of currents flowing out of it.

Byteasar now wants to install ammeters on the wires, which will determine the intensity of the currents. Of course, he could install an ammeter on each cable, but the operation is lengthy, tedious, and of course, very expensive. Thus Byteasar reckons how to reduce costs to a minimum — its well known that it is possible to determine the current flowing in certain wires based on the measurements in others.

There is a certain cost of ammeter installation at each of the wires. This cost can be positive (it is necessary to burn precious fuel and devote an ammeter unit) or negative (at some places Byteasar expects to find ~~beautiful Vicugnian ladies~~ precious minerals, which will recompense his losses). Find a way to install ammeters (in other words — an adequate set of wires), which has the lowest possible total cost, and allows to determine the current levels in all the wires.

Input

The first line contains two integers n and m ($2 \le n \le 200\,000$, $1 \le m \le 500\,000$) describing the number of nodes and the number of wires. In the next m lines are descriptions of the wires: j-th line contains three integers a_j, b_j, c_j ($1 \le a_j, b_j \le n$, $a_j \ne b_j$, $-10^9 \le c_j \le 10^9$) indicating that the j-th wire connects nodes a_j and b_j, and the cost of installation of the ammeter on it amounts to c_j. Each pair of nodes is connected by not more than one wire.

Output

The only line of output should contain one integer — the minimum cost of an adequate set of ammeters.

Examples

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

Problem C. The Feast

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

King Byteasar plans to organize a great feast on the occasion of his 128th birthday. He decided to seat his guests at round tables in such a way that nobody is alone at the table. Has already chosen n persons he could invite to the party. Then he ranked the guests in order, starting from the most important one, and has numbered them sequentially from 1 to n (most important is the guest number 1).

The guests are very demanding: each of them informed the king about their wishes regarding the person who can sit next to their right. The King wants everyone to be seated in good company, so he will not allow that the requirements of the guests are not met. It may therefore be so that it will not be possible to invite all n people. Byteasar asked you (King's court computer specialist) to work out the best out of possible sets of invited guests, and the exemplary seating arrangement at round tables. The King, when asked what did he mean by a best set of guests, surprisingly soberly replied, for 127-year-old computer layman:

To compare the two sets of guests, I choose the person with the lowest number, which belongs to exactly one of the compared sets. This particular person belongs to a better set.

With such a specifically described order, you can actually clearly identify which of the potential sets of guests is the best.

Input

The first line of input contains an integer n ($2 \leq n \leq 2000$) denoting the number of guests. i-th of consecutive n lines contains preferences description of the person numbered i. Preference description begins with integer k_i ($k_i \geq 0$). It is followed by k_i pairwise distinct numbers of people — these are integers ranging from 1 to n, different from i. Any such number describes one person who can sit to the right of the person number i. The sum of the numbers k_i does not exceed 5000.

Output

In the first line of output an integer s should be written, denoting the number of tables in the worked out solution. Next s lines should contain descriptions of the individual tables. Description of a table starts with an integer g denoting the number of guests seated at it. Next follow g numbers designating their numbers. Guests numbers should be presented sequentially, in counter-clockwise order. The first of the presented guests will be sitting on the right to the last guest.

Examples

standard input	standard output
6 3 2 6 3 0 1 4 1 1 1 4 1 5	1 3 1 3 4

In the above example, it is better to invite person 1, 3 and 4 to the party rather than 1, 4, 5 and 6. According to the King's criterion a better set is represented by the person number 3.

Problem D. The Furniture

Input file:	*standard input*
Output file:	*standard output*
Time limit:	6 seconds
Memory limit:	512 mebibytes

Byteasar plans to decorate his new flat. For this purpose he has gone to the nearby `BITKEA` chain shop and bought n types of furniture, to be more precise: c_i pieces of furniture type i.

Assembling the first piece of i type furniture (including Byteswedish language instruction analysis) will take him a_i minutes. Putting together subsequent furniture units makes Byteasar more experienced at it — assembling second and each subsequent piece of i furniture type would take him about d_i minutes less than the previous unit of this type.

Byteasar decided that he will assembly a certain number of furniture pieces today. For a number of queries, each specifying a number m_i, he would like to know what is the shortest time to assembly certain m_i units of purchased furniture.

Input

The first line of input contains two integers n and k ($1 \leq n, k \leq 500$) indicating the number of types of furniture and the number of queries. i-th of n following lines contains three integers a_i, d_i, c_i ($1 \leq a_i, d_i, c_i \leq 10^9, a_i > (c_i - 1) \cdot d_i$), which are a description of i-th type of the purchased furniture. The i-th of k successive lines contains the integer m_i ($1 \leq m_i \leq 20\,000$).

Output

Output should consist of k lines. The i-th line should contain the minimum number of minutes required to assemble m_i furniture pieces. You can assume that the assembly of certain m_i furniture units will always be possible.

Example

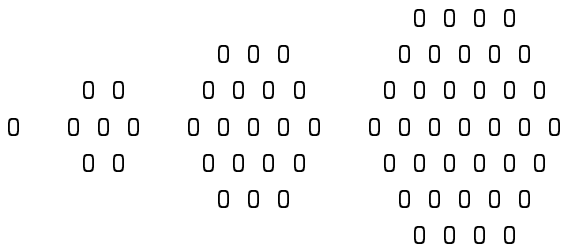
standard input	standard output
3 6 20 3 6 25 20 2 19 1 19 1 2 3 4 5 6	19 30 49 62 70 75

Problem E. Hexagons

Input file:	*standard input*
Output file:	*standard output*
Time limit:	10 seconds
Memory limit:	512 mebibytes

When Bytebara was a little girl, she liked to arrange cubic toy blocks into squares. She took a certain number of blocks and tried to divide them into minimum number of squares. She was always able to do it in such a manner, that the number of squares was at most four.

Today Bytebara is an adult and earns trillions of Bytethalers, instead of playing toy blocks. Recently she has read that every natural number is a sum of at most four squares. In addition, each number is the sum of at most three triangular numbers ($\frac{n(n+1)}{2}$ form number). It evoked her with her childhood fun and she began to play with coins. But instead of arranging coins in squares or triangles, she began to arrange them in hexagons.



For a given number of coins Bytebara wants to know, what the smallest number of hexagons it could be divided into. For example, 27 coins can be divided into three hexagons ($27 = 1 + 7 + 19$).

Input

Input consists of T tests ($T \le 1000$). For $1 \le i \le T$, the i-th line of input contains the number K_i ($1 \le K_i \le 10^{12}$) — number of coins, being at Bytebara's disposal.

$(T + 1)$-th line of input contains number 0, indicating the end of input.

Output

For each of T numbers of coins, output the minimum number of hexagons, which it can be divided into.

Example

standard input	standard output
1 6 7 19 27 0	1 6 1 1 3

Problem F. The Mine

Input file:	*standard input*
Output file:	*standard output*
Time limit:	8 seconds
Memory limit:	512 mebibytes

All Byteotian diamond deposits occur at the same depth and are in the form of rectangles with sides parallel to the axes of a certain coordinate system. The deposits do not overlap, but may touch by their sides or by their vertices.

Geologists have not yet been able to explain the reason for this phenomenon, but developed a precise map of the deposits. This is connected with plans of constructing new opencast mine. Excavation pit (also rectangular in shape) will be l meters in length and w meters wide. Various sites are being considered for its location. For each possible location calculate the area of the deposits operated by a given mine, that is, the area of the deposits located exactly below the excavation pit of the mine.

Input

The first line of input contains two integers n and q ($1 \leq n, q \leq 100\,000$) indicating the number of Byteotian diamond fields and the number of possible mine locations. The second line contains two integers l and w ($1 \leq l, w \leq 1\,000\,000$) — the length and the width of the excavation pit of the mine. Each of the following n lines contains four integers: a_i, b_i, c_i, d_i. (a_i, b_i) are the coordinates of the bottom left, and (c_i, d_i) — of the upper right corner of i-th diamond deposit field. The areas of the fields are positive numbers. All the coordinates belong to the range $[0, 1\,000\,000]$.

Following are q lines, each containing a pair of non-negative integers x_i, y_i ($x_i + l, y_i + w \leq 1\,000\,000$). These are considered locations of the lower left corner of the excavation pit (then the upper right corner of the excavation pit has the following coordinates $(x_i + l, y_i + w)$).

Output

Your program should produce exactly q lines, each containing one integer s_i — the area of diamond deposits covered by the i-th position of the excavation field.

Examples

standard input	standard output
4 3 4 2 1 1 3 3 0 5 1 6 2 4 3 6 1 6 4 7 1 1 0 4 1 5	4 3 4

Problem G. Nim3

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

The three brothers Antek, Bartek and Cezary really like playing Nim. At some point, they came up with the idea to play this game between the three of them. At the beginning they were lining up many pebble stacks. Then they moved in turns, taking any amount of pebbles (a positive number, obviously) from the chosen stack. Antek, the youngest of the brothers, always performed the first move, the next move belonged to Bartek, another to Cazary, then again moved Antek, and so on. Loser was the player who could not make a move. Unfortunately, it was usually Antek.

His older brothers did not compete with each other and did everything to make him lose. The youngest of the brothers decided to change it and suggested that the first place be taken by a person who will perform the last move; third — by a player who cannot move, and the second place — by the last of the three brothers. Brothers analysed the new rules and came to the conclusion that in this new game the way in which a move is performed will be guided by the following algorithm defined recursively:

1. Consider all the moves that you can do and for each of them calculate what will be the result of the game (applying the algorithm recursively).
2. Select the move which gives the best result (in the case of many equally good opportunities select any one of them).

Knowing the initial situation, decide who will win the game in case all three brothers play in line with the above description.

Input

The first line of input contains an integer t ($1 \leq t \leq 1000$) denoting the number of test cases. Subsequent t lines describe one test case each.

A single test case description starts with a positive integer n_i, which indicates the number of stacks of pebbles in the game. Next follows a string of n_i integers a_{ij} ($1 \leq a_{ij} \leq 10^{18}$), which determines the initial number of pebbles in each stack. The sum of all n_i does not exceed 1 000 000.

Output

The output should consist of t lines with answers for subsequent test cases.

The answer for one case is the letter A, B, C, which is the first letter of the winner's name of the corresponding game.

Examples

standard input	standard output
3 2 5 6 3 2 2 2 4 1 2 3 4	B C A

Problem H. The Sorting Machine

Input file:	*standard input*
Output file:	*standard output*
Time limit:	4 seconds
Memory limit:	512 mebibytes

Byteasar works at the Sorting Machines Institute. For several months he has been busy inventing new and efficiently working machine, and finally he succeeded: the invention devised by Byteasar can easily sort any sequence consisting of not more than k different natural numbers.

Byteasar's boss approached him today with an urgent order to sort a certain permutation of numbers ranging from 1 to n, where $n > k$. Byteasar quickly realised that the permutation is too long for his machine, however he failed to persuade the boss to drop this idea. Presently he had just enough time left to run the machine once, selecting a consistent k-element fragment of boss's permutation. He decided to do it in such a way, that the resulting permutation is lexicographically as small as possible. But how to choose the right fragment to perform the sorting?

Input

The first line contains two integers n and k ($2 \leq k < n \leq 1\,000\,000$), indicating the length of the boss's permutation and the maximum fragment length, which is still possible to be sorted by Byteasar's machine. The second line contains a permutation of the numbers $\{1, \ldots, n\}$, i.e. a sequence of n pairwise distinct integers ranging from 1 to n.

Output

The first and only line of output should contain a sequence of n numbers denoting the minimum permutation Byteasar can obtain running his machine not more than once.

Examples

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

Problem I. Space Mission

Input file:	*standard input*
Output file:	*standard output*
Time limit:	6 seconds
Memory limit:	512 mebibytes

Byteland is preparing to launch its first rocket into space. Byteasar is one of the space program employees and he is responsible for the process of boarding the rocket by the astronauts. The interior of the rocket consists of n cabins connected by two-way passages in such a manner that we can move between each two cabins in exactly one way (if we do not turn back). Crossing each corridor takes one Byteotian second. Cabins are numbered from 1 to n. The entrance to the rocket leads directly into the cabin number 1.

There will be n astronauts boarding the rocket, also numbered from 1 to n. For each $1 \leq i \leq n$ astronaut number i will take quarters in cabin numbered i. Astronauts enter the rocket one after the other in one second intervals (Byteotian seconds) and use the shortest way to reach their cabins. Astronaut number i starts to unpack his gear after reaching his cabin, which takes him exactly a_i Byteotian seconds.

The order of residents boarding the rocket must be such, that no one should have to go through the cabin, which is already occupied by its resident (regardless whether that occupant has just finished unpacking, or not).

Byteasar's task is to plan rocket boarding process to run as quickly as possible, meaning that between the first astronaut entering the rocket and the moment at which all astronauts finish unpacking, the least amount of time passes.

Input

The first line of input contains an integer n ($2 \leq n \leq 500\,000$) denoting the number of astronauts and the number of cabins. The second line contains a sequence of n integers a_i ($1 \leq a_i \leq 10^9$). Number a_i determines how much time astronaut number i needs to unpack. Subsequent $n-1$ lines describe the rocket's cabin system. Each of them contains two integers a and b ($1 \leq a, b \leq n$), which indicate that cabins numbered a and b have direct corridor connection.

Output

Output the minimum time required by all astronauts to board and accommodate in their cabins.

Examples

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

Problem J. Turtles

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

Jack and Chip came across the herd of cold stiff turtles. Turtles form a row, from left to right, and some are standing on their feet, some are lying on their shells. It is too cold for the turtles to move — even those standing on their feet cannot escape.

Seeing this, the boys came up with an idea to play a game. Jack will start, and then they will be making moves in turns. Every move consists of turning a certain number of turtles (at least one) to the other side, there are however certain limitations:

- The first (left-most) reversed turtle has to stand on his feet and is reversed onto the shell.
- The turtles that are reversed must be positioned equally apart.

The winner is the player who manages to put all the turtles on their shells, in his move.

Help Jack to determine, whether can win this game, and if so, what move he should make.

Input

The only line of input contains a string of letters describing the initial state of the turtles, from left to right. The letter `N` indicates that the turtle is standing on his feet, and `S` that lies on its shell.

Not all turtles initially lie on their shells.

The number of letters shall not be less than 1 and not more than 3000.

Output

In case the game can be won by Jack, list the status of the turtles after one of Jack's winning moves, in the same format as at the input. In case Jack cannot win, one word must be produced: `NIE` (Polish for no).

Examples

standard input	standard output
NSNNN	NSNNS
NSNNS	NIE