Problem A. Dwarf Tower

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	256 Mebibytes

Little Vasya is playing a new game named "Dwarf Tower". In this game there are n different items, which you can put on your dwarf character. Items are numbered from 1 to n. Vasya wants to get the item with number 1.

There are two ways to obtain an item:

- You can buy an item. The i-th item costs c_i money.
- You can craft an item. This game supports only m types of crafting. To craft an item, you give two particular different items and get another one as a result.

Help Vasya to spend the least amount of money to get the item number 1.

Input

The first line of input contains two integers n and m ($1 \le n \le 10\,000; 0 \le m \le 100\,000$) — the number of different items and the number of crafting types.

The second line contains n integers c_i — values of the items ($0 \le c_i \le 10^9$).

The following m lines describe crafting types, each line contains three distinct integers a_i, x_i, y_i — a_i is the item that can be crafted from items x_i and y_i ($1 \le a_i, x_i, y_i \le n$; $a_i \ne x_i$; $x_i \ne y_i$; $y_i \ne a_i$).

Output

The output should contain a single integer — the least amount of money to spend.

Examples

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

Problem B. Ray Distance

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	64 mebibytes

You are given two rays which describe by pair of points. Your have to find the distance between rays.

Input

Input contains four pairs of integers — the descriptions of the first and of the second rays. Each ray describes by coordinates of its begin and by coordinates of some point at this ray. All numbers in input don't exceed 100000 by absolute value.

Output

Output the answer with absolute or relative error not greater than $1e-6$.

Examples

standard input	standard output
0 0 1 0 0 1 0 2	1.0000000000

Problem C. Interesting Equation

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

Given two positive integers A and B such as $GCD(A, B) = 1$. Find minimal positive integer N such as $N/A = X^A$ and $N/B = Y^B$, where X and Y are positive integers.

Input

First line of the input contains one integer T — number of test cases ($1 \le T \le 10^4$). Each of next T lines contains two integers A abd B ($2 \le A, B \le 10^4$, $GCD(A, B) = 1$).

Output

For each test case print in the new line the answer modulo $10^9 + 7$.

Example

standard input	standard output
4 2 3 9 2 9 10 10000 9999	648 4608 573245882 907923460

Problem D. Race

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	64 mebibytes

At the time 0 n racers are placed in the points $x_1, x_2, \ldots, x_n$ from the starting line.

Each racer runs with the constant speed $v_1, v_2, \ldots, v_n$. All racers are running in one direction.

You need find positive time t, when distance between current first and last place is minimal possible.

Input

First line of the input contains one integer n — number of the racers.

Each of next n lines contains two integers: x_i — distance from the starting line to the i-th racer at the time 0 and v_i — speed of this racer, respectively ($0 \le x_i \le 10^7$, $0 \le v_i \le 10^7$).

Output

Print two numbers: t — time in seconds passed from the time 0 to the moment when the distance between first and last place become minimal and l — value of this minimum. Print answers with absolute or relative error 10^6 or better.

Example

standard input	standard output
3 0 40 30 1 40 30	1.0 30.0
5 90 100 100 70 100 70 110 60 120 35	0.5000 5.0

Problem E. Spies

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

N spies received some important information, each spy got her own piece of information. As their boss, you can

- Send two spies to meet each other and exchange all information they got (directly or on previous meetings). For each two spies you know the price you must pay to organize this meeting.
- Send a spy to the action. To ensure the success, each piece of information must be known to atleast one spy sent to the action. For each spy you know amount of money you spend for sending her to action.

Calculate minimum budget needed to ensure the success of the action.

Input

First line of the input contains one positive integer N — number of spies ($2 \leqslant N \leqslant 1000$). Each of next N lines contains N positive integers, does not exceedint 10^6. An integer in k-th row and m-th column denotes price you must pay to organize meeting of k-th and m-th spies. It is guaranteed that it coincides with integer in m-th row and k-th column; if $k = m$ then it is equal to zero.

Next line contains N positive integers; i-th of those integers denotes price you must pay to send the i-th spy to the action.

Output

Print one integer — minimub budget of the action.

Example

standard input	standard output
3 0 6 9 6 0 4 9 4 0 7 7 7	17
3 0 17 20 17 0 10 20 10 0 15 9 12	34
5 0 3 12 15 11 3 0 14 3 20 12 14 0 11 7 15 3 11 0 15 11 20 7 15 0 5 10 10 10 10	28

Problem F. Bacon's Cypher

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	64 mebibytes

Programmer Vasya was down on his luck. Instead of a vacation, he was sent to a scientific conference.

"It is necessary to increase your competence," his boss said, "it's an important conference on cryptography, and it's held in France, where they used encryption in the days of de Richelieu and cracked codes in the days of Viete."

One of the talks at the conference was about the attempts to solve Bacon's cyphers. The speaker proposed a hypothesis that the key to Bacon's secrets could be found if all possible substrings of Bacon's works were analyzed.

"But there are too many of them!" Vasya expressed his astonishment.

"Not as many as you think," the speaker answered, "count them all and you'll see it yourself."

That evening Vasya found on the Web the complete set of Bacon's works. He wrote a program that converted the texts into one long string by removing all linebreaks, spaces, and punctuation marks. And now Vasya is confused because he doesn't know how to calculate the number of different substrings of this string.

Input

You are given a nonempty string consisting of lowercase English letters. The string is no longer than 5000 symbols.

Output

Output the number of different substrings of this string.

Examples

standard input	standard output
aaba	8

Problem G. Hentium Scheduling

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	256 megabytes

John is developing the new operating system Jindox Lion that will run on systems with new Hentium processor that has asymmetric dual-core support. There are two cores in Hentium, one core has fast memory access, another one has faster arithmetics. So scheduling memory-dependent processes on the first core and calculation intensive processes on the second one, one can take advantage of the architecture.

John is working on scheduling system boot processes on the two cores. There are n system initialization processes in Jindox. John has measured that the i-th process will take a_i nanoseconds to execute on the first core or b_i nanoseconds to execute on the second core. However, processes, scheduled at different cores are slower to communicate, so if the i-th process is scheduled at one core and the j-th process is scheduled at another one, it will take additional $c_{i,j}$ nanoseconds for them to communicate with each other.

Since John wants his system to be as stable as possible, he doesn't want processes at different cores to execute simultaneously. Therefore there is a special semaphore that ensures that only one process at one of the cores is executed at each moment of time. Now John needs to arrange some processes to the first core, and others to the second, so that the total time of their execution and communication was as small as possible. Help him to do it.

Input

The input file contains several test cases.

The first line of each test case contains n — the number of processes ($2 \le n \le 100$). The following n lines describe processes, each line contains two integers: a_i and b_i — execution times at the first core and at the second core, respectively ($1 \le a_i, b_i \le 10^9$). The following n lines contain n integers each, the j-th number of the i-th line is $c_{i,j}$ ($0 \le c_{i,j} \le 10^9$, $c_{i,i} = 0$, $c_{i,j} = c_{j,i}$).

Input is followed by $n = 0$, it must not be processed. The sum of n for all test cases doesn't exceed 1000.

Output

For each test case print two lines. The first line must contain one integer: the total time of execution and communication of processes in case of optimal scheduling, in nanoseconds.

The second line must contain n integers: for each process print 1 if it must be executed on the first core, or 2 if it must be executed on the second core. If there are several optimal solutions, output any one.

Examples

standard input	standard output
4 1 8 2 5 10 1 15 2 0 2 0 0 2 0 10 4 0 10 0 1 0 4 1 0 0	11 1 2 2 2

Problem H. Memory Manager

Input file:	*standard input*
Output file:	*standard output*
Time limit:	0.5.second
Memory limit:	64 mebibytes

Bob needs to design a memory manager for the new standard library of the H++ language. The manager might use an array of N memory cells numbered from 1 to N. The manager is receiving requests from applications to allocate or deallocate memory.

An allocation request (K) means that an application requests to allocate K contiguous memory cells. If the manager has a free block of K contiguous cells, it allocates the memory. The memory is allocated as the leftmost segment of length K from the longest of all free blocks. If there are several free blocks of maximum length, the one with the minimum address of the leftmost cell is chosen. The allocated cells are marked as occupied and cannot be used for allocation until they are deallocated. If there is no block of K contiguous cells, the request is declined.

Deallocation request (T) means that manager has to deallocate the memory which was allocated by the T-th request starting from 1. It is guaranteed that request number T is an allocation request and that there was no deallocation request for this request before. Deallocated memory can be used again for further memory allocation. If request number T has been declined, current deallocation request should be ignored. Write a memory manager simulation that meets all the requirements listed above.

Input

The first line of the input contains two numbers N and M, number of memory cells and number of requests respectively. ($1 \le N \le 2^{31} - 1$, $1 \le M \le 10^5$). Each of the next M lines contains one integer. The $(i + 1)$-th line contains a positive integer K if the i-th request is a request to allocate K memory cells ($1 \le K \le N$) or a negative integer $-T$ if the i-th request is request to deallocate memory allocated on request number T ($1 \le T < i$).

Output

For each allocation request, output its result on a separate line. If memory has been allocated, output the number of the first cell in the allocated block, otherwise output -1.

Example

standard input	standard output
6 8 2 3 -1 3 3 -5 2 2	1 3 -1 -1 1 -1

Problem I. Heavy Chain Clusterization

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	256 Mebibytes

A group of biologists is trying to find a cure for a viral disease. They have tried many antibodies of various origins that could potentially fight the viral antigens, and have selected n antibodies that seem to work best during experiments.

Each antibody is identified by its *heavy chain* — a sequence of amino acids.

The set of antibodies form a *similarity cluster*, if at least one of the following holds:

- k-prefixes (first k amino acids) of all their heavy chains are equal;
- k-suffixes (last k amino acids) of all their heavy chains are equal.

In order to simplify the future research, biologists want to group antibodies to similarity clusters.

You need to split the given antibodies to a minimum number of similarity clusters.

Input

The first line contains two integers n and k — the number of heavy chains and the length of sequence of amino acids to coincide ($1 \le n \le 5\,000$, $1 \le k \le 550$).

The following n lines contain sequences of amino acids that form heavy chains of antibodies. Each amino acid described with an uppercase English letter. Each heavy chain contains at least k and no more than 550 amino acids.

Output

The first line of output must contain a single integer — the minimum number of similarity clusters. The following lines must contain descriptions of clusters, one per line.

Each description starts with m_i — the number of antibodies in the cluster and is followed by m_i integers — numbers of these antibodies. Antibodies are numbered in the order of appearance in the input starting from one.

Each antibody must be present in exactly one cluster.

Examples

standard input	standard output
4 1 AA AB BB BA	2 2 1 2 2 3 4
3 2 ABA BAB XY	3 1 1 1 2 1 3

Problem J. Rectangles

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

Let there be N nondegenerate rectangles with sides parallel to coordinate axes. It is known that the borders of these rectangles do not intersect and there are no identical borders. Let us say that rectangle A is *nested* inside rectangle B if any point belonging to A also belongs to B, and $A = B$. Let us say that rectangle A is *immediately nested* inside rectangle B if A is nested inside B and A is not nested inside any other rectangle that is nested inside B. Write a program to analyse immediate nesting of rectangles.

Input

The first line of the input file specifies an integer N . Then N lines follow. Of these lines, line number i specifies 4 integers: x_{i1} , y_{i1}, x_{i2}, y_{i2} — the coordinates of the opposite vertices of the i-th rectangle ($1 \le N \le 10^5$, $0 \le x_{ij}, y_{ij} \le 10^9$).

Output

Write N lines to the output file. Line number i must contain the number of a rectangle inside which rectangle number i is immediately nested, or 0 if there is no such rectangle in the input set.

Example

standard input	standard output
5 0 0 10 10 1 2 4 5 2 3 3 4 7 7 8 8 0 11 1 12	0 1 2 1 0

Problem K. Street Directions

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	256 mebibytes

According to the Automobile Collision Monitor (ACM), most fatal traffic accidents occur on two-way streets. In order to reduce the number of fatalities caused by traffic accidents, the mayor wants to convert as many streets as possible into one-way streets. You have been hired to perform this conversion, so that from each intersection, it is possible for a motorist to drive to all the other intersections following some route.

You will be given a list of streets (all two-way) of the city. Each street connects two intersections, and does not go through an intersection. At most four streets meet at each intersection, and there is at most one street connecting any pair of intersections. It is possible for an intersection to be the end point of only one street. You may assume that it is possible for a motorist to drive from each destination to any other destination when every street is a two-way street.

Input

The input consists of a number of cases. The first line of each case contains two integers n and m. The number of intersections is n ($2 \leq n \leq 1000$), and the number of streets is m. The next m lines contain the intersections incident to each of the m streets. The intersections are numbered from 1 to n, and each street is listed once. If the pair (i, j) is present, (j, i) will not be present. End of input is indicated by $n = m = 0$.

Output

For each case, print the case number (starting from 1) followed by a blank line. Next, print on separate lines each street as the pair "i j" to indicate that the street has been assigned the direction going from intersection i to intersection j. For a street that cannot be converted into a one-way street, print both "i j" and "j i" on two different lines. The list of streets can be printed in any order. Terminate each case with a line containing a single "#" (without quotes) character.

Note: There may be many possible direction assignments satisfying the requirements. Any such assignment is acceptable.

Example

standard input	standard output
7 10 1 2 1 3 2 4 3 4 4 5 4 6 5 7 6 7 2 5 3 6 0 0	1 1 2 2 4 3 1 3 6 4 3 5 2 5 4 6 4 6 7 7 5 #

Problem L. Internet Shop

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

In the Aweson internet shop, the following payment rules apply. All purchases are paid by thalers, the internal Aweson currency. When a new customer is registered in the system, he gets a bonus account, initially with 0 thalers. At the moment of each purchase, a customer has a choice: either to use available bonuses or to accumulate them. When using available bonuses, a customer may pay for any part of the purchase with bonus thalers, and the remaining part of the cost, if any, is paid with regular thalers. When accumulating bonuses, a customer may not use existing bonus thalers, but exactly 1% of the cost is added to the bonus account.

Mihail made a plan of purchases on Aweson. Now he wants to know which choice to make for each purchase and how to pay for them so that the total number of regular thalers he spends is minimum possible.

Input

The first line of input contains an integer n: the number of purchases ($1 \le n \le 100$). The second line contains n integers p_1, p_2, ..., p_n separated by single spaces: the cost of the first, second, ..., n-th purchase in thalers ($0 < p_i \le 10^5$, all p_i divide evenly by 100).

Output

On the first line, print one integer s: the minimum total amount of regular thalers with which Mihail can make all his purchases in the given order.

On the next n lines, print a strategy which allows to spend exactly s regular thalers. Each of these lines must contain two integers c_k and b_k separated by a «plus» sign surrounded with single spaces: how much regular thalers and how much bonus thalers Mihail should spend on k-th purchase. Obviously, these numbers must be nonnegative, and their sum must be equal to the cost of k-th purchase. If $b_k = 0$, then k-th purchase is accumulating bonuses, and if $b_k > 0$, it is using bonuses. Naturally, during each purchase, Mihail can not spend any more bonuses than he currently has, and the sum of all c_k must be exactly s.

The number of bonus thalers after all n purchases is irrelevant. If there are several strategies with the minimum possible s, print any one of them.

Examples

standard input	standard output
2 100 100	199 100 + 0 99 + 1
3 100 10000 100	10100 100 + 0 10000 + 0 0 + 100

Explanations

In the first example, Mihail gets one bonus thaler from the first purchase. Then we spend it to pay for a part of the second purchase.

In the second example, Mihail also gets one bonus thaler from the first purchase. However, it is not profitable to spend it on the second purchase. Instead, Mihail can get 100 more thalers from the second

purchase. After that, he can fully pay for the third purchase with bonus thalers, and even have one bonus thaler left.

Problem M. Electricity

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	256 mebibytes

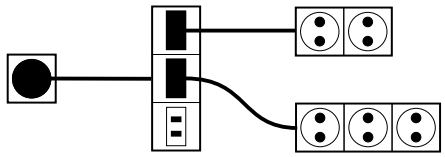
You are managing a power network for a programming competition and you have to connect a lot of computers to the power supply. Unfortunately, there are two standards for electrical plug and socket: A and B. These standards are incompatible, so the plug of standard A can only be plugged in the socket of standard A and the plug of standard B can only be plugged in the socket of standard B.

In the main competition hall, there is exactly one main socket of type A. Every computer that will be used in this programming competition has one plug of type A. Thereby only one computer can be connected directly to the main socket. But you have a number of power strips of two types.

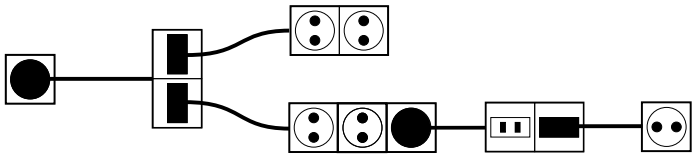
- Power strips of the first type have one plug of type A and several sockets of type B.
- Power strips of the second type have one plug of type B and several sockets of type A.

All the power strips are very powerful and can withstand any load. So you can create a power network by connecting one power strip of the first type to the main socket, then some power strips of the second type to this power strip of the first type, etc. At the end you will get several available sockets of type A for computers.

Your task is to find the maximum number of computers that can be connected to the power network, using available power strips.



Possible solution for the first example.



Possible solution for the second example.

Input

The first line of input file contains two integer numbers n and m — the number of power strips of the first and the second type ($0 \le n, m \le 100\,000$).

The second line contains n integer numbers a_i — the number of sockets on the power strips of the first type ($1 \le a_i \le 1000$).

The second line contains m integer numbers b_i — the number of sockets on the power strips of the second type ($1 \le b_i \le 1000$).

Output

Output the maximum number of computers that can be connected to the power network.

Examples

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

Problem N. Piatra Neamt

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	256 mebibytes

After becoming the ruler of Piatra Neamt, PAM began a long stream of reforms. Under her wise guidance, the Piatra Neamt metropole turned into an empire, and PAM proclaimed herself Empress.

As the Empire grew, PAM realised that a good network of roads is needed. Because PAM likes trees very much, PAM decided that her road network will resemble a tree. More specifically, when the road network was being built, PAM made sure the network had the following properties:

- Unoriented-ness: All the roads are bidirectional.
- Connectivity: Regardless of which city a person is in, the person must be able to reach any city of the Empire.
- Cycle-less-ness: There should never be more than one path between two cities.

The empire continued to grow, and PAM realized that some cities are very far away from Piatra Neamt, the capital of the Empire. Because of the great distance of remote cities to the capital, collecting taxes required a very long period of time.

Although she loves her native city, for efficiency reasons, PAM has decided to move the capital from Piatra Neamt to another city that would optimize tax collection.

PAM decided to attach a cost to each city and choose the city with the lowest cost as the capital. PAM decides that the cost of a city is the sum of the distances from that city to all the other cities of the Empire.

Because her Empire has a LOT of cities (we won't tell you how many, but you can rest assured it is at most 100 000), PAM has asked you, the Empire programmer, to find the cities that are the best candidates.

Input

The first line of the input contains N, the number of cities in the Empire. The following $N-1$ lines each contain a pair of integers representing cities that are connected by a direct road. Cities are numbered from 1 to N, and all the roads have equal length.

Output

The first line of the output must contain the minimum cost, the number of candidates (i. e. cities that have the minimum cost), and the candidates. The list of candidates must be given in ascending order. All the numbers must be separated by spaces.

Example

standard input	standard output
2 1 2	1 2 1 2