Problem A. Max Flow Min Cost

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

You are given the directed graph, each edge of which is assigned to capacity and cost. Find maximum flow of minimum cost from vertex 1 to vertex n.

Input

First line of input contains integers n and m ($2 \le n \le 100$, $0 \le m \le 1000$), the number of vertices and the number of edges respectively.

Each of the following m lines contains four integers: 1-based vertex indices connected by an edge, its capacity and its cost.

Capacities and costs do not exceed 10^5.

It is guaranteed that there are no negative cycles in the graph.

Output

Output the only integer, which is the cost of the maximum flow of minimum cost from 1 to n.

Examples

standard input	standard output
4 5 1 2 1 2 1 3 2 2 3 2 1 1 2 4 2 1 3 4 2 3	12

Problem B. Assignments

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

Given an integer matrix C of size $n \times n$. You need to select n elements such as each row and each column will contain exactly one selected element and the sum of values of selected elements is minimal possible.

Input

First line of the input contains one integer n ($2 \le n \le 300$). Each of next n lines contains n integers C_{ij}. All integers in the input are non-negative and does not exceed 10^6.

Output

To the first line of the output print one integer — a minimal sum.

Each of next n lines must contain two integers — number of row and column of the selected cell.

If there are several solutions, print any of them.

Examples

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

Problem C. Beer

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

Everyone knows that World Finals of ACM ICPC 2004 were held in Prague. Besides its greatest architecture and culture, Prague is world famous for its beer. Though drinking too much is probably not good for contestants, many teams took advantage of tasting greatest beer for really low prices.

A new beer producing company *Drink Anywhere* is planning to distribute its product in several of the n European cities. The brewery is located near Prague, that we would certainly call city number 1. For delivering beer to other cities, the company is planning to use logistics company *Drive Anywhere* that provides m routes for carrying goods. Each route is described by the cities it connects (products can be transported in either direction), its capacity — the number of barrels of beer that can be transported along this route each day, and the cost of transporting one barrel of beer along it. To deliver beer to some city it may be necessary (or advantageous) to use several routes consequently, and maybe even deliver beer using several different paths.

Each city is in turn characterized by the price that local pubs and restaurants are ready to pay for one barrel of beer. You may assume that demand for beer is essentially unlimited in each city, since this is the product that will always find its consum

Drink Anywhere is not planning to distribute its beer in Prague for a while, because of the high competition there, so it is just planning to provide beer to other cities for now. Help it to find out, what is the maximal income per day it can get.

Input

The first line of the input file contains n and m — the number of cities in Europe we consider and the number of delivery routes respectively ($2 \le n \le 100$, $1 \le m \le 2000$). The next line contains $n-1$ integer numbers — prices of a barrel of beer in European cities $2, 3 \ldots, n$ respectively (prices are positive integers and do not exceded 1000).

The following m lines contain four integer numbers each and describe delivery routes. Each route is specified by the numbers of cities it connects, its capacity, and the price of transporting one barrel of beer along it (the capacity and the price are positive integers, they do not exceed 1000).

Output

Output the maximal income the company can get each day.

Examples

standard input	standard output
4 4 80 50 130 1 2 80 50 2 4 40 90 3 1 40 60 3 4 30 50	3000

Problem D. Domino in Casino

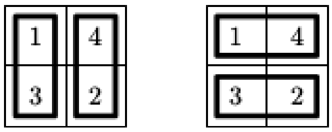
Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	256 mebibytes

Domino is well known as a game played by people at streets when they relax after a workday. So it was, until recently John Bigbuck introduced domino in his casino “*BUMP*” (Bring Us Money, Please).

Of course, ordinary domino games are not well suited for casino, so John had to introduce his own game. The game is played on a rectangular board of size $m \times n$. Each cell of the board contains some integer number.

The player has k domino tiles — rectangles 2×1. The player puts the tiles to the board without overlapping and his winning is calculated as the sum of products of numbers under each tile.

For example, there are 2 ways to put 2 tiles on a 2×2 board. For the board below, the better way to put the tiles is shown on the left — in this case the sum is $1 \times 3 + 4 \times 2 = 11$. If the player puts tiles to the board as shown on the right picture, the sum would be $1 \times 4 + 3 \times 2 = 10$, which is smaller.



Given the board, and the number of the tiles the player has, find what is the maximal sum he can get.

Input

The first line of the input file contains integer numbers m, n and k ($1 \le m \le 16$, $1 \le n \le 100$, $1 \le k \le 200$). The following m lines contain n integer numbers each and describe board. Numbers written on the board are non-negative and do not exceed 1000. It is guaranteed that it is possible to arrange all tiles on a board.

Output

Output one integer — the maximal sum a player can get.

Examples

standard input	standard output
2 2 2 1 4 3 2	11

Problem E. Automata Programming

Input file:	*standard input*
Output file:	*standard output*
Time limit:	5 seconds
Memory limit:	256 mebibytes

There are n tasks, each of which is described with three integers: s_i, task start time; t_i, task duration and c_i, task profit. You have k executors, each of which is capable of performing at most 1 task at the moment. Each task may be performed by at most executor. Executor should perform task from its start to its finish completely in order to gain profit. Switching from one task to another takes 0 time, i.e. it is possible for an executor to move from task i to task j iff $s_i + t_i \le s_j$.

What is the most profitable set of tasks that may be executed by given k executors (i.e. the set of task with maximum total cost)?

Input

First line contains two integers n and k ($1 \le n \le 1000$, $1 \le k \le 50$), the number of tasks and executors, respectively.

Each of the following n lines contains three integers s_i, t_i, c_i ($1 \le s_i, t_i \le 10^9$, $1 \le c_i \le 10^6$), where s_i is the start time of the task i, t_i is its duration and c_i is its profit.

Output

Print n integers $x_1, x_2, \ldots, x_n$, where x_i should be 1 if task i should be performed by some executor and 0 otherwise. If there are several optimum solutions, print any of them.

Examples

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

Problem F. Inspection

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

You are in charge of a team that inspects a new ski resort. A ski resort is situated on several mountains and consists of a number of slopes. Slopes are connected with each other, forking and joining. A map of the ski resort is represented as an acyclic directed graph. Nodes of the graph represent different points in ski resort and edges of the graph represent slopes between the points, with the direction of edges going downwards.

Input

The first line of the input file contains a single integer number n ($2 \le n \le 100$) — the number of points in the ski resort. The following n lines of the input file describe each point of the ski resort numbered from 1 to n. Each line starts with a single integer number m_i ($0 \le m_i < n$ for i from 1 to n) and is followed by m_i integer numbers a_{ij} separated by spaces. All a_{ij} are distinct for each i and each a_{ij} ($1 \le a_{ij} \le n$, $a_{ij} \ne i$) represents a slope going downwards from point i to point a_{ij}. Each point in the resort has at least one slope connected to it.

Output

On the first line of the output file write a single integer number k — the minimal number of helicopter flights that are needed to inspect all slopes. Then write k lines that describe inspection routes for each helicopter flight. Each route shall start with single integer number from 1 to n — the number of the drop off point for the helicopter flight, followed by the numbers of points that will be visited during inspection in the corresponding order as the slopes are inspected going downwards. Numbers on a line shall be separated by spaces. You can write routes in any order.

Example

standard input	standard output
8 1 3 1 7 2 4 5 1 8 1 8 0 2 6 5 0	4 1 3 4 8 2 7 5 8 3 5 7 6

Problem G. Binary Tree on Plane

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3 seconds
Memory limit:	256 mebibytes

A root tree is a directed acyclic graph that contains one node (root), from which there is exactly one path to any other node.

A root tree is binary if each node has at most two outgoing arcs.

When a binary tree is painted on the plane, all arcs should be directed from top to bottom. That is, each arc going from u to v must meet the condition $y_u > y_v$.

You've been given the coordinates of all tree nodes. Your task is to connect these nodes by arcs so as to get the binary root tree and make the total length of the arcs minimum. All arcs of the built tree must be directed from top to bottom.

Input

The first line contains a single integer n ($2 \le n \le 400$) — the number of nodes in the tree. Then follow n lines, two integers per line: x_i, y_i ($\|x_i\|, \|y_i\| \le 10^3$) — coordinates of the nodes. It is guaranteed that all points are distinct.

Output

If it is impossible to build a binary root tree on the given points, print "`-1`". Otherwise, print a single real number — the total length of the arcs in the minimum binary tree. The answer will be considered correct if the absolute or relative error doesn't exceed 10^{-6}.

Examples

standard input	standard output
3 0 0 1 0 2 1	3.650281539872885
4 0 0 1 0 2 1 2 0	-1

Problem H. Aarelia Mountains

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

You are a powerful wizard travelling across the Aarelia kingdom. Your path lies across the land of steep hills and mountains. The mountain area can be represented as a row of n regions; each region has its *elevation* h_i.

Unfortunately, you have no climbing equipment (and, frankly, no climbing skills), and your Flight spells are exhausted. However, you will be able to travel across the area when region elevations will form a non-decreasing sequence, that is, $h_i \le h_{i+1}$ for each i from 1 to $(n-1)$ (you are travelling from right to left, and high falls are no threat for you).

The only spell that can help you is the Earthquake spell. The spell affects several adjacent regions and allows you to alter their elevation. You know m varieties of the Earthquake spell; each spell has its range l_i (that is, number of adjacent regions affected by the spell), its energy cost c_i and can either raise or lower all affected regions' elevation by 1 (a single spell can be applied either to raise or lower regions, but not both). Each spell can be applied to any range of length l_i that lies completely within the area, and can be cast an unlimited number of times (paying energy cost after each repetition). It is possible that some of the regions will have negative elevation after casting a spell.

Determine whether it is possible to alter the landscape so that you may travel safely across the area, and if it is, find the lowest total energy you have to spend on casting the spells.

Input

The first line contains two space-separated integers n and m ($1 \le n, m \le 200$).

The second line contains n space-separated integers h_i — initial elevation of the regions ($0 \le h_i \le 10^6$).

The next m lines describe varieties of the Earthquake spell you know. The i-th of these lines describes the spell variety according to format "t_i l_i c_i"; here t_i is the character "+" if the spell allows to raise elevations by 1, or "-" if it allows to lower elevations by 1; l_i and c_i are the range and the energy cost of the spell, respectively ($1 \le l_i \le n$, $1 \le c_i \le 10^6$).

Output

It it is possible to travel across the area, print the lowest total energy that must be spent on casting the spells, otherwise print -1.

Examples

standard input	standard output
3 2 3 2 1 + 1 1 - 1 1	2
3 1 3 2 1 + 2 1	-1

Problem I. Reserve

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

Protoss prepare a new attack. They have a lot of battle units to attack, but it is necessary to leave some Zealots (it is one of the most useful Protoss battle unit) on the base (as a reserve) to be sure that the base is protected from any counterattack. One Zealot requires c_{ij} minerals to defend the base if employed from day i to day j. Attack operation proceeds for n days. Protoss commander Tryt o'Kill calculated that in day k he needs at least b_k Zealots on the base ($k = 1, 2, \ldots, n$). The general can keep exactly b_k units on the base and pay them required number of minerals, but he wants to minimize the amount of minerals spent during all n days. The general can also keep more than b_k units on day k if this will result in smaller total amount of minerals.

Input

First line of the input file contains number of days n ($1 \le n \le 50$).

Next n lines describe the cost of keeping Zealots. So, j-th number in the $(i+1)-th$ line of the input file is $c_{i,i+j-1}$. All costs are non-negative and don't exceed 10 000. The last line contains b_k. Consider $0 \le b_k \le 2000$.

All numbers in the input are integral.

Output

In the first line write total amount of minerals General has to pay to Zealots to defend the base. Output a triangular table: j-th number of the $(i+1)$-th line should contain the number of Zealots which have to defend the base from day i to day $i+j-1$. This number shouldn't exceed 10^9. Keep in mind that Protoss General wants to minimize amount of minerals paid to Zealots. It is guaranteed that the optimal amount of minerals doesn't exceed 10^9.

Examples

standard input	standard output
3 3 2 3 3 2 3 2 3 2	7 0 1 1 0 1 0

Problem J. Longest Shortest Path

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

You are given a directed graph and two nodes s and t. The given graph may contain multiple edges between the same node pair but not self loops. Each edge e has its initial length d_e and the cost c_e. You can extend an edge by paying a cost. Formally, it costs $x \cdot c_e$ to change the length of an edge e from d_e to $d_e + x$. (Note that x can be a non-integer). Edges cannot be shortened.

Your task is to maximize the length of the shortest path from node s to node t by lengthening some edges within cost P. You can assume that there is at least one path from s to t.

Input

The first line of the input contains five integers N, M, P, s, and t: N $(2 \leq N \leq 200)$ and M $(1 \leq M \leq 2,000)$ are the number of the nodes and the edges of the given graph respectively, P $(0 \leq P \leq 10^6)$ is the cost limit that you can pay, and s and t $(1 \leq s, t \leq N,\ s \neq t)$ are the start and the end node of objective path respectively. Each of the following M lines contains four integers v_i, u_i, d_i and c_i, which mean there is an edge from v_i to u_i $(1 \leq v_i, u_i \leq N,$ $v_i \neq u_i)$ with the initial length d_i $(1 \leq d_i \leq 10)$ and the cost c_i $(1 \leq c_i \leq 10)$.

Output

Output the maximum length of the shortest path from node s to node t by lengthening some edges within cost P. The output can contain an absolute or a relative error no more than 10^{-6}.

Examples

standard input	standard output
3 2 3 1 3 1 2 2 1 2 3 1 2	6
3 3 2 1 3 1 2 1 1 2 3 1 1 1 3 1 1	2.5000000
3 4 5 1 3 1 2 1 2 2 3 1 1 1 3 3 2 1 3 4 1	4.25