Problem Set 7 — Network Flow — 15-295 Fall 2019

A. Network Flow (Easy)

time limit per test: 2 seconds
memory limit per test: 1024 megabytes

Find the maximum flow through a graph from the source to the sink. You may not assume anything about the graph (there may be loops, there may be multiple edges between two vertices, etc)

(Any max flow algorithm should pass this problem)

Input

The first line will contain two integers, n ($2 \le n \le 100$) and m ($1 \le m \le 1\,000$), denoting the number of nodes and the number of edges.

The second line will contain two integers s and t ($1 \le s, t \le n$) denoting the source and sink ($s \ne t$).

The next m lines contain 3 integers u, v ($1 \le u, v \le n$) and c ($0 \le c \le 10\,000$) denoting an edge in the graph.

Output

Output the maximum flow of the graph from s to t.

Examples

input	output
2 1 1 2 1 2 3	3

input	output
4 5 1 4 1 2 10 1 3 10 2 3 1 2 4 10 3 4 10	20

Problem B Those are not the droids you're looking for

Time limit: 1 second
Memory limit: 64 MB

> Bar owner: Hey. We don't serve their kind here.
> Luke: What?
> Bar owner: Your droids – they'll have to wait outside. We don't want them here.

Planet Tatooine is quite far from the center of the Galaxy. It is at the intersection of many hyperspace paths and it hosts smugglers and hoods of all sorts. The pilots who visited external territories have been to the space port bar called Mos Eisley for a drink at least once.

In this bar you can find a bunch of rascals and scoundrels from all over the Galaxy. The bar owner is ready to make drinks for any client except for, perhaps, a droid. Usually the bar has a lot of smugglers hanging out there. Each smuggler spends at least a minutes inside hoping to meet a good client. Cargo owners show up quite often as well. They usually find a dealer quickly, so they never spend more than b minutes in the bar.

The imperial stormtroopers are searching through Tatooine for the escaped droids. The bar owner said that no droids had ever been on his territory. He also said that nobody except for smugglers and cargo owners had been in the place recently.

Help the stormtroopers find out if the owner is a liar. For that, you are going to need the daily records from the sensor on the entrance door. The sensor keeps record of the time when somebody entered the bar or left it. The stormtroopers got the records after the bar had been closed, so there was nobody in the bar before or after the sensor took the records. You can assume that the sensor is working properly. That is, if somebody went through the bar door, the sensor made a record of that. You can also assume that the bar clients go in and out only through the door with the sensor. But the bar owner and the staff use the 'staff only' door.

Input

The first line of the input contains integers a and b ($1 \le a, b \le 10^9, b + 1 < a$). The second line contains integer n the number of records from the sensor ($2 \le n \le 1000$). The i th of the next n lines contains two integers t_i and d_i ($1 \le t_i \le 10^9, d_i \in \{0, 1\}$) the time of the i th record and direction (0 in the bar, 1 out of the bar). The records in the input are listed in the increasing order of t_i.

Output

If there is no doubt that somebody who was neither a smuggler nor a cargo owner visited the bar, print `"Liar"` on a single line. Otherwise, print a line `"No reason"`. And in the following lines list information on all visitors of the bar. The information about a visitor consists of two space separated numbers the time this visitor entered the bar and the time he left the bar. If there are multiple solutions that correspond to the sensor records, print any of them.

Examples

input	output
6 3 4 1 0 2 0 5 1 10 1	No reason 1 10 2 5
6 3 4 1 0 2 0 6 1 10 1	Liar

Problem **C**
Flame of Nucleus

Input: nucleus.in, Output: nucleus.out, Timelimit: 30sec.

Year 20XX — a nuclear explosion has burned the world. Half the people on the planet have died. Fearful.

One city, fortunately, was not directly damaged by the explosion. This city consists of N domes (numbered 1 through N inclusive) and M bidirectional transportation pipelines connecting the domes. In dome i, P_i citizens reside currently and there is a nuclear shelter capable of protecting K_i citizens. Also, it takes D_i days to travel from one end to the other end of pipeline i.

It has been turned out that this city will be polluted by nuclear radiation in L days after today. So each citizen should take refuge in some shelter, possibly traveling through the pipelines. Citizens will be dead if they reach their shelters in exactly or more than L days.

How many citizens can survive at most in this situation?

Input

The input consists of multiple test cases. Each test case begins with a line consisting of three integers N, M and L ($1 \leq N \leq 100$, M ≥ 0, $1 \leq L \leq 10000$). This is followed by M lines describing the configuration of transportation pipelines connecting the domes. The i-th line contains three integers A_i, B_i and D_i ($1 \leq A_i < B_i \leq N$, $1 \leq D_i \leq 10000$), denoting that the i-th pipeline connects dome A_i and B_i. There is at most one pipeline between any pair of domes. Finally, there come two lines, each consisting of N integers. The first gives $P_1, \ldots, P_N$ ($0 \leq P_i \leq 10^6$) and the second gives $K_1, \ldots, K_N$ ($0 \leq K_i \leq 10^6$).

The input is terminated by EOF. All integers in each line are separated by a whitespace.

Output

For each test case, print in a line the maximum number of people who can survive.

Sample Input

```
1 0 1
51
50
2 1 1
1 2 1
1000 0
0 1000
4 3 5
1 2 4
1 3 1
3 4 2
0 1000 1000 1000
3000 0 0 0
```

Output for the Sample Input

```
50
0
3000
```

D. Array and Operations

time limit per test: 1 second
memory limit per test: 256 megabytes

You have written on a piece of paper an array of n positive integers $a[1], a[2], ..., a[n]$ and m *good* pairs of integers $(i_1, j_1), (i_2, j_2), ..., (i_m, j_m)$. Each *good* pair (i_k, j_k) meets the following conditions: $i_k + j_k$ is an odd number and $1 \le i_k < j_k \le n$.

In one operation you can perform a sequence of actions:

- take one of the *good* pairs (i_k, j_k) and some integer v ($v > 1$), which divides both numbers $a[i_k]$ and $a[j_k]$;
- divide both numbers by v, i. e. perform the assignments: $a[i_k] = \frac{a[i_k]}{v}$ and $a[j_k] = \frac{a[j_k]}{v}$.

Determine the maximum number of operations you can sequentially perform on the given array. Note that one pair may be used several times in the described operations.

Input

The first line contains two space-separated integers n, m ($2 \le n \le 100$, $1 \le m \le 100$).

The second line contains n space-separated integers $a[1], a[2], ..., a[n]$ ($1 \le a[i] \le 10^9$) — the description of the array.

The following m lines contain the description of *good* pairs. The k-th line contains two space-separated integers i_k, j_k ($1 \le i_k < j_k \le n$, $i_k + j_k$ is an odd number).

It is guaranteed that all the *good* pairs are distinct.

Output

Output the answer for the problem.

Examples

input	output
3 2 8 3 8 1 2 2 3	0

input	output
3 2 8 12 8 1 2 2 3	2

E. Fox And Dinner

time limit per test: 2 seconds
memory limit per test: 256 megabytes

Fox Ciel is participating in a party in Prime Kingdom. There are n foxes there (include Fox Ciel). The i-th fox is a_i years old.

They will have dinner around some round tables. You want to distribute foxes such that:

1. Each fox is sitting at some table.
2. Each table has at least 3 foxes sitting around it.
3. The sum of ages of any two adjacent foxes around each table should be a prime number.

If k foxes f_1, f_2, ..., f_k are sitting around table in clockwise order, then for $1 \le i \le k$ - 1: f_i and f_{i+1} are adjacent, and f_1 and f_k are also adjacent.

If it is possible to distribute the foxes in the desired manner, find out a way to do that.

Input

The first line contains single integer n ($3 \le n \le 200$): the number of foxes in this party.

The second line contains n integers a_i ($2 \le a_i \le 10^4$).

Output

If it is impossible to do this, output "Impossible".

Otherwise, in the first line output an integer m ($1 \le m \le \frac{n}{3}$): the number of tables.

Then output m lines, each line should start with an integer k -=– the number of foxes around that table, and then k numbers — indices of fox sitting around that table in clockwise order.

If there are several possible arrangements, output any of them.

Examples

input	output
4 3 4 8 9	1 4 1 2 4 3

input	output
5 2 2 2 2 2	Impossible

input	output
12 2 3 4 5 6 7 8 9 10 11 12 13	1 12 1 2 3 6 5 12 9 8 7 10 11 4

input	output
24 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25	3 6 1 2 3 6 5 4 10 7 8 9 12 15 14 13 16 11 10 8 17 18 23 22 19 20 21 24

Note

In example 1, they can sit around one table, their ages are: 3-8-9-4, adjacent sums are: 11, 17, 13 and 7, all those integers are primes.

In example 2, it is not possible: the sum of 2+2 = 4 is not a prime number.

F Pool construction

You are working for the International Company for Pool Construction, a construction company which specializes in building swimming pools. A new client wants to build several new pool areas.

A pool area is a rectangular grid of $w \times h$ square patches, consisting of zero or more (possibly disconnected) pools. A pool consists of one or multiple connected hole patches, which will later be filled with water. In the beginning, you start with a piece of land where each patch is either a hole in the ground ('.') or flat grass ('#'). In order to transform this land into a pool area, you must adhere to the following:

- You can leave a patch as it is. This costs nothing.
- If the patch is grass in the beginning, you can dig a hole there. This costs d EUR.
- If the patch is a hole in the beginning, you can fill the hole and put grass on top. This costs f EUR.
- You *must* place special boundary elements along each edge running between a final grass patch and a final hole patch, to ensure that water does not leak from the pool. This costs b EUR per boundary element.
- The outermost rows and columns of the pool area must always be grass.

You are given the task of calculating the cost of the cheapest possible pool area given the layout of the existing piece of land.

Input

On the first line a positive integer: the number of test cases, at most 100. After that per test case:

- one line with two integers w and h ($2 \leq w, h \leq 50$): the width and height of the building site.
- one line with three integers d, f and b ($1 \leq d, f, b \leq 10\,000$): the costs for digging a new hole, filling an existing hole, and building a boundary element between a pool and grass patch.
- h lines of w characters each, denoting the layout of the original building site.

Output

Per test case:

- one line with an integer: the cost of building the cheapest possible pool area from the original piece of land.

Sample in- and output

Input	Output
3 3 3 5 5 1 #.# #.# ### 5 4 1 8 1 #..## ##.## #.#.# ##### 2 2 27 11 11 #. .#	9 27 22