# y2018-4-2. Потоки

## A. Maximum Flow

5 seconds, 1024 megabytes

There is a system of nodes and pipes that are used for water transportation. For each pipe you know the maximum speed that water flowing through that pipe can achieve. Water flows through pipes in such a way that for every node (except for the source and the sink), the volume of water going into that node is the same as the volume of water going out of that node.

Your task is to find the maximum volume of water that can flow from the source to the sink, and the speed of water inside each pipe. Pipes are bidirectional, meaning that water can flow in any direction. There can be more than one pipe between each pair of nodes.

### Input

The first line contains integer $N$ — the number of nodes ($2 \le N \le 100$). The source has number $1$ and the sink has number $N$. The second line contains integer $M$ ($1 \le M \le 5000$) — number of pipes in the system. Next $M$ lines describe pipes. Each pipe is describe by three integers $A_i$, $B_i$, $C_i$, where $A_i$, $B_i$ are the nodes connected by this pipe ($A_i \ne B_i$), and $C_i$ ($0 \le C_i \le 10^4$) — the maximum speed for that pipe.

### Output

The first line should contain the maximum volume of water that can flow from the source to the sink. Then print $M$ lines, the speed of water for the corresponding pipe. If the direction of water flow is not equal to the input order, print the speed with the minus sign.

**input**

```
2
2
1 2 1
2 1 3
```

**output**

```
4
1
-3
```

## B. Cut

2 seconds, 1024 megabytes

Given an undirected graph, find the minimum cut between vertices 1 and $n$.

### Input

The first line contains $n$ ($2 \le n \le 100$) — the number of vertices and $m$ ($0 \le m \le 400$) — the number of edges. Next $m$ lines describe edges. One edge is described by two vertices it connects and its capacity (an integer less than or equal to $10\,000\,000$). No two vertices are connected by more than one edge.

### Output

Output the number of edges in the minimum cut and the total capacity of all edges in the first line. Then, print the indices of edges in the minimum cut in increasing order (edges are numbered in the same way as they are listed in the input).

**input**

```
3 3
1 2 3
1 3 5
3 2 7
```

**output**

```
2 8
1 2
```

Statement is not available on English language

## C. Улиточки

2 секунды, 1024 мегабайта

Две улиточки Маша и Петя сейчас находятся в на лужайке с абрикосами и хотят добраться до своего домика. Лужайки пронумерованы числами от 1 до $n$ и соединены дорожками (может быть несколько дорожек соединяющих две лужайки, могут быть дорожки, соединяющие лужайку с собой же). По соображениям гигиены, если по дорожке проползла улиточка, то вторая по той же дорожке уже ползти не может. Помогите Пете и Маше добраться до домика.

### Входные данные

В первой строке файла записаны четыре целых числа — $n$, $m$, $s$ и $t$ (количество лужаек, количество дорог, номер лужайки с абрикосами и номер домика). В следующих $m$ строках записаны пары чисел. Пара чисел $(x, y)$ означает, что есть дорожка с лужайки $x$ до лужайки $y$ (из-за особенностей улиток и местности дорожки односторонние). Ограничения: $2 \leq n \leq 10^5, 0 \leq m \leq 10^5, s \neq t$.

### Выходные данные

Если существует решение, то выведите YES и на двух отдельных строчках сначала последовательность лужаек для Машеньки (дам нужно пропускать вперед), затем путь для Пети. Если решения не существует, выведите NO. Если решений несколько, выведите любое.

**входные данные**

```
3 3 1 3
1 2
1 3
2 3
```

**выходные данные**

```
YES
1 3
1 2 3
```

Дан орграф, найти два непересекающихся по ребрам пути из $s$ в $t$, вывести вершины найденных путей.

## D. The Great Wall

2 seconds, 1024 megabytes

King Louis has two sons. They dislike each other, and the King is worried that after his demise the country will be ruined by wars. That's way Louis decided to split the country into two parts, each one being ruled by one of the sons. He prepared a thrones for each of them in cities $A$ and $B$, and wants to build the minimum possible number of fragments of the wall such that there's not path from $A$ to $B$.

The country has be represented as a $m \times n$ grid. Some cells contain mountains, other cells can be easily traversed. Moreover, some cells can be turned into the wall fragment, and some cannot. It's possible to move from one cell to any of the four adjacent cells, if it doesn't contain a wall segment or a mountain.

### Input

The first line contains $m$ and $n$ ($1 \le m, n \le 50$). Next $m$ lines have $n$ each and describe the map of the country. Characters have the following meaning: «#» — mountain, «.» — a cell, that can be used for building a wall, «-» — a cell that cannot be used for building a wall, «A» and «B» — cities $A$ and $B$.

### Output

Print the minimum number of wall segments $F$ in the first line. The, print the map using the same format as in the input. Use «+» to denote built walls.

If building walls is impossible, print $-1$.

**input**

```
5 5
--...
A-.#-
.#.#-
--.--
--.-B
```

**output**

```
3
--+..
A-+#-
+#.#-
--.--
--.-B
```

**input**

```
1 2
AB
```

**output**

```
-1
```

**input**

```
2 2
A#
#B
```

**output**

```
0
A#
#B
```

## E. Bring Them There

2 seconds, 1024 megabytes

By the year 3141, the human civilization has spread all over the galaxy. The special hypertunnels are used to travel from one star system to another. To use the hypertunnel, you fly to a special location near the source star using your spaceship, activate the hyperjumper, fly through the hypertunnel, get out near your destination star and fly to the planet you need. The whole process takes exactly one day. A small drawback of the system is that for each tunnel every day only one spaceship can travel using this tunnel.

You are working in the transportation department of the "Intergalaxy Business Machines" company. This morning your boss has assigned a new task to you. To run the programming contest IBM needs to deliver $K$ supercomputers from Earth where the company headquarters are located to the planet Eisiem. Since supercomputers are very large, one needs the whole spaceship to carry each supercomputer. You are asked to find a plan to deliver the supercomputers that takes as few days as possible. Since IBM is a very powerful corporation, you may assume that any time you need some tunnel for hyperjump, it is at your service. However, you still can use each tunnel only once a day.

### Input

The first line of the input file contains $N$ — the number of star systems in the galaxy, $M$ — the number of tunnels, $K$ — the number of supercomputers to be delivered, $S$ — the number of the solar system (the system where planet Earth is) and $T$ — the number of the star system where planet Eisiem is ($\{2 \le N \le 50\}$, $\{1 \le M \le 200\}$, $\{1 \le K \le 50\}$, $\{1 \le S, T \le N\}$, $\{S \ne T\}$).

Next $M$ lines contain two different integer numbers each and describe tunnels. For each tunnel the numbers of star systems that it connects are given. The tunnel can be traveled in both directions, but remember that each day only one ship can travel through it, in particular, two ships cannot simultaneously travel through the same tunnel in opposite directions. No tunnel connects a star to itself and any two stars are connected by at most one tunnel.

### Output

On the first line of the output file print $L$ — the fewest number of days needed to deliver $K$ supercomputers from star system $S$ to star system $T$ using hypertunnels. Next $L$ lines must describe the process. Each line must start with $C_i$ — the number of ships that travel from one system to another this day. $C_i$ pairs of integer numbers must follow, pair $A, B$ means that the ship number $A$ travels from its current star system to star system $B$.

It is guaranteed that there is a way to travel from star system $S$ to star system $T$.

**input**

```
6 7 4 1 6
1 2
2 3
3 5
5 6
1 4
4 6
4 3
```

**output**

```
4
2  3 2  4 4
3  2 4  3 3  4 6
3  1 4  2 6  3 5
2  1 6  3 6
```

## F. Group tournament

2 seconds, 1024 megabytes

In our capitalist-run, dog-eat-dog world money is everything, and big sports are no exception. All participating teams have already bought enough points for the next season, and all the local hockey federation has to do now is distribute the results of the upcoming games. However, some teams felt generous and apart from buying points also bought the results of some matches. Initially the federation officials thought it would only make their life easier: the more games are fixed, the less work. It was only later that they understood their wrong and asked us to be a part of their scheme and help them distribute the results of the games in the upcoming season.

The local hockey tournament follows a round scheme: $N$ teams participate, each team plays a game against each and every other team strictly once. Teams score points for games according to the following rules:

- If the winning team is defined at the end of the regulation time of the match, it scores 3 points, and the other team gets none.
- If a game is tied after regulation time, overtime ensues. In this case the winner gets 2 points and the loser gets 1. The overtime is unlimited and lasts until someone scores a goal.

Based on the tournament results, a team's score is calculated as the sum of its points earned in all games played.

### Input

The first line of the input file contains an integer $N$ — the number of tournament participants ($2 \le N \le 100$). Teams are numbered from 1 to $N$.

The following $N$ lines of the file each contain $N$ symbols and are in essence a tournament table for the given moment of time.

The symbol $a_{ij}$ in the line $i$ and position $j$ denotes the result of a game scheduled to be played by team number $i$ against team number $j$ ($1 \le i, j \le N$). It can be one of the following:

- `'W'` — means that the team $i$ will win a match against $j$ in the regulation time
- `'w'` — the team $i$ will win in a match against $j$ in the overtime
- `'l'` — the team $i$ will lose to the team $j$ in the overtime
- `'L'` — the team $i$ will lose to the team $j$ in the regulation time
- `'.'` — if the result of the game between $i$ and $j$ is not yet determined
- `'#'` — if $i$ equals $j$, it means that there is no such game, i.e. a team cannot play against itself.

It is guaranteed that the table is correct. More formally:

- $a_{ij}$ = `'#'` for all $i = j$
- if $a_{ij}$ = `'.'`, then $a_{ji}$ = `'.'`
- $a_{ij}$ = `'W'` when and only when $a_{ji}$ = `'L'`
- $a_{ij}$ = `'w'` when and only when $a_{ji}$ = `'l'`

The last line of the input file contains $N$ integers $p_i$ — the number of points the $i$-th team must score ($1 \leq i \leq N$).

### Output

The output file must contain a completely filled tournament table in the same format as that in the input file.

It is guaranteed that a solution exists. If there are several solutions, print any of them.

input

```
4
#..W
.#w.
.l#.
L..#
8 6 3 1
```

output

```
#wWW
l#wW
Ll#w
LLl#
```