DISCRETE STRUCTURES FOR COMPUTER SCIENCE
(CS F222)

ASSIGNMENT-2

CONTACT DETAILS:

- VAMSI NALLAPAREDDY 2017A7PS0018H
- VARUN GUMMA 2017A7PS0165H

A. A New World

Ash, Dawn and Brock are on their way to the Spear Pillar, and have just entered *Mt. Coronet*. At the Spear Pillar, Team Galactic's head, Cyrus, has captured Dialga and Palkia and has begun the process to create a new universe, but our trio is unable to make it there on time as they have lost their way in the darkness.

Mt. Coronet has a few parts. The entrance and Spear Pillar are located in different parts of *Mt. Coronet*. Some of the parts of *Mt. Coronet* are connected by bidirectional tunnels, while some aren't. Can you help our heroes find the path with *minimal tunnels* to the Spear Pillar from the entrance and stop Cyrus?

Input

The first line has two integers, N ($2 \le N \le 2000$) and M ($1 \le M \le \frac{N(N-1)}{2}$), the number of parts (which are consecutively numbered from 1 to N), and the number of tunnels in *Mt. Coronet*.

The next M lines have two space-separated integers U, V ($1 \le U, V \le N$, $U \ne V$) denoting a bidirectional tunnel between the U^{th} and V^{th} part. The last line has two space-separated integers, S and D ($1 \le S, D \le N$, $D \ne S$), denoting the part in which the entrance and Spear Pillar are located respectively. It is guaranteed that there are no multiple tunnels between any two parts.

Output

If no path exists between the entrance and Spear Pillar, print -1; else, in the first line print L, the path length, and in the following line print L space separated integers A_1 , A_2 , ..., A_L ($1 \le A_i \le N$), denoting the path from the entrance to the spear pillar where $A_1 = S$ and $A_L = D$. If there are multiple answers, print any one of them.

input

```
11 9
2 1
1 3
6 7
7 8
9 7
10 4
5 1
1 4
4 11
2 10
```

output

```
3
2 1 4 10
```

input

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

output

```
-1
```

B. Virus Attack

Elzo D' Plaza, a famous Computer Hacker, has set his eyes on the new start-up, Foodle. The Foodle workstation has a few computers, a few of which are interconnected. Elzo has installed his secret virus program in some of the workstation computers at Foodle. His virus program can *creep* from one system to another, if they are interconnected. Over the night, Foodle is devastated. The engineers at Foodle were able to figure out the PCs in which Elzo had initially installed the virus. Now can you help them figure out how many systems are still intact?

Input

The first line has three space-separated integers, N ($1 \le N \le 2000$) denoting the number of systems at Foodle, M ($1 \le M \le \frac{N(N-1)}{2}$) the number of pairs of interconnected computers at Foodle and P ($1 \le P \le N$) the number of systems in which Elzo had initially installed the virus. The next M lines have two integers U and V ($1 \le U, V \le N, U \ne V$) denoting an interconnection between system U and system V. The last line has P space-separated integers $A_1, A_2, \ldots, A_P$ ($1 \le A_i \le N$) denoting the system numbers in which Elzo installed the virus. It is assured that there are no multiple connections between the same two PCs.

Output

In the first line, output L, the number of unaffected computers, and in the next line print $B_1, B_2, \ldots, B_L$ ($1 \le B_j \le N$) representing the list of unaffected computers, if any exist, else print *Black Out*.

---

input

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

output

```
3
5 6 7
```

---

input

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

output

```
Black Out
```

C. Friendship Goals

Freshmen year at MIT has just begun and students want to make as as many new friends as they can. Slowly, a few groups started emerging. A group is formed is formed in the following way:
Say A, B are two friends, then the current group is just AB. Say, now A befriends C, the new group is ABC. Further, if B befriends Q and C befriends X, the new group will be ABCXQ.

As a Sophomore student at MIT, you are curious about the new groups forming on campus. As you have the knowledge of who befriends whom, can you quickly jot down a list of all groups on campus? *A single student (who has not befriended anyone) is also considered a separate group.*

Input

The first line contains two space-separated integers, N ($1 \le N \le 26$), M ($1 \le M \le 325$) denoting the number of students who joined MIT this year (these will be N consecutive characters starting from A), and the number of friendships that have happened over the time. The following M lines have two space separated characters, X and Y indicating X befriends Y (and Y befriends X).

Output

In the first line print L, the number of group in campus, and in the next L lines, print the groups. (See the sample case for better understanding). *You can output the groups in any order. Further, the members of the group can be printed in any order.*

---

input

```
10 6
B E
G C
I F
C B
A E
F J
```

output

```
4
ABCEG
FIJ
D
H
```

---

input

```
5 5
B D
C D
A E
A B
C E
```

output

```
1
ABCDE
```

D. Wallas's Challenge

You are at the final gym of Pokemon Sapphire fan-version and are ready to challenge the gym leader Wallas, but there lies a different challenge ahead of you. As you enter the gym, you are suddenly materialized onto the (x1, y1)th tile of a N×N ice grid. Your goal is to reach the gym leader Wallas who is on the (x2, y2)th tile of the grid after covering (walking over) all the tiles on the grid *exactly once*. On the ice grid, you can either take a single step forward, backward, to the left or the right, if and only if the tile you are moving to was initially unvisited, and within the boundaries.

As you're suspicious that you might not always be able to reach Wallas by covering all tiles *exactly once*, so you decide to write a program that will aid you to solve the problem.

Input

The first line has a single integer N ($1 \le N \le 7$), the size of the grid. The next two lines have two space separated integers (x1, y1), (x2, y2) respectively ($0 \le x1, x2, y1, y2 < N$), denoting the initial grid tile you land on, and the final tile you plan to reach after *walking over all tiles exactly once*. It is guaranteed that the pairs (x1, y1) and (x2, y2) will be distinct.

Output

If "YES" (without quotes), if you can reach Wallas from (x1, y1) by covering all tiles *exactly once*, else print "NO" (without quotes).

---

input
```
2
0 0
0 1
```

output
```
YES
```

---

input
```
5
0 0
4 4
```

output
```
YES
```

---

input
```
5
1 1
3 4
```

output
```
NO
```

E. Find the Words

Sheldon Cooper gives a quick puzzle to Leonard on their way to work. He gives him a square character matrix and asks him to find the number of ways of forming a particular word from the matrix. To form a word, Leonard can start any point in the matrix and can move to an adjacent cell (cells that share a corner or an edge). Each cell can be used exactly once while forming one word. Note that, those cells can obviously be reused for forming other words. As Leonard is too busy driving, he asks you, Howard, to code this question so as to pacify Sheldon's tantrums.

Input

The first line has one integer N ($1 \leq N \leq 500$), denoting the size of the N×N character grid. The next N lines have have N strings, having N uppercase characters each, denoting the character grid. The next line has a single integer M ($1 \leq M \leq N^2$), denoting the length of the string we need to find.The final line has the string S ($|S| = M$), whose occurences have to found.

Output

Output a single integer denoting the number of occurences of the given word in the matrix.

---

input

```
7
SHELDON
HSTYUPQ
EHGXBAJ
LMNNQQI
DTYUIOP
OZXCVBN
NQWERTY
7
SHELDON
```

output

```
4
```

---

input

```
4
ABCD
ABCD
ABCD
ABCD
2
BC
```

output

```
10
```

F. Colour it!

The Pokemon game developers at Nintendo are planning the new region, Galar. As they created the town map of Galar, they realized that the graph formed using the cities as vertices and the roads as edges might be bipartite, but they are unsure about it. As they are short on time, and the deadline to release the game is round the corner, can you help them figure out if the graph so formed is indeed a bipartite? If it is, can you suggest a colouring scheme for it, such that no two immediately connected cities are of the same colour ?

Input

The first line has two space-seperated integers N ($1 \le N \le 500$), M ($1 \le M \le \frac{N(N-1)}{2}$) denoting the number of cities in Galar and the number of bidirectional roads between the cities. The next M lines have two space-separated integers U and V ($1 \le U, V \le N$) denoting a bidirectional road between cities U and V.

Output

If the graph is a not a bipartite, print “NO”, else print “YES” in the first line and in the second line, print the colour scheme, (use R to denote first colour and B to denote second colour), where the i^{th} character stands for the colour assigned to the i^{th} city.

---

input

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

output

```
NO
```

---

input

```
4 4
2 3
3 1
4 1
2 4
```

output

```
YES
RRBB
```

G. Resource Management

A kind of directed graph used by operating system to check for deadlocks is a *Wait-for* graph. In this kind of graph, if a process P_i is waiting for a process P_j, a single directed edge between P_i and P_j is incorporated (here the process are treated as vertices). A deadlock scenario arises if a cycle is formed with the given nodes and directed edges. Further, the nodes that are lead to atleast one node that is involved in the deadlock are also deadlocked i.e. can't progess. Given a *Wait-for* graph, can you find all the nodes (processes) that are deadlocked?

Input

The first line has two space-seperated integers N ($0 \le N \le 20$), M ($1 \le M \le \frac{N(N-1)}{2}$) denoting the number of processes in the system, and the number of dependencies (waits) in the system. The next M lines have two space-separated process U, V ($1 \le U, V \le N$), denoting process_U is waiting for process_V.

Output

If no deadlocks are present in the scenario, print "No Deadlocks", else print "Deadlock" and all the processes that are deadlocked. The deadlocked processes can be printed in any order. (see the sample case for better understanding).

---

input

```
5 5
0 1
2 3
3 4
1 2
```

output

```
Deadlock
0 1 2 3
```

explanation

In this case, P_1, P_2, P_3, form a cycle and hence, a deadlock is present. As P_0 is waiting for P_1 (which is deadlocked), even P_0 is deadlocked.

---

input

```
5 4
0 1
0 2
1 3
1 4
```

output

```
No Deadlock
```

H. Butterfly Effect

In his attempt to undo all changes in the timelines, Okabe Rintaro has begun his work on the Time Leap machine. But he realizes that he may not be able to jump between a few timelines using this device. As his research continues, he faces a shocking revelation.

He realizes that if he visits a partiular timeline (say T_p) before another another timeline (say T_q), he might end up creating a butterfly effect, and wishes to avoid it at all costs. As he is quite caught up with his work on the Time Leap machine, he asks you, Super Hacker Daru, to figure out a path to the Steins;Gate Worldline without encountering two known timelines in the path.

Input

The first line has three space-seperated integers N ($1 \le N \le 3000$), M ($1 \le M \le \frac{N(N-1)}{2}$), S ($1 \le S \le N$) denoting the number of timelines, the number of unidirectional leaps Okabe can perform between those timelines and the final timeline (Steins;Gate Worldline) Okabe has to reach. The next M lines have two space-separated integers U and V ($1 \le U, V \le N$) denoting that Okabe can jump from timeline U to timeline V. The last line has two space-separated integers P, Q ($1 \le P, Q \le N$) denoting Okabe cannot visit timeline Q after timeline P during the leaps. Assume that Okabe is always initially in timeline 1.

Output

If Okabe can reach the Steins;Gate worldline from his current timeline without visiting timeline Q after timeline P, print “YES” followed by the path can take, else print “NO”. If multiple paths are present which satisfy the condition, print any one.

input
```
6 6 4
1 2
2 3
3 5
5 4
6 5
1 6
3 5
```
output
```
YES
1 6 5 4
```

input
```
6 6 6
1 2
3 5
4 5
2 4
3 1
4 6
1 4
```
output
```
NO
```

I. Ticket Cost

You have been selected for the International Informatics Olympiad and must travel to New York from Delhi. There are flights between various cities, and you must plan your flights accordingly, as you have very limited money. Plan your trip so that you spend minimum money on your tickets.

Input

The first line has four space-separated integers N ($1 \le N \le 300$) the number of cities, M ($1 \le M \le \frac{N(N-1)}{2}$), the number of bidirectional flights between those cities, P the airport number for Delhi and Q ($1 \le P, Q \le N, Q \ne P$) the airport number for New York. The next M lines have three space-separated integers, U, V and C ($1 \le U, V \le N, 1 \le C \le 10^5$) with C representing the cost of the flight between cities U and V or V to U.

Output

In the first line, print the minimum fare of your travel, and in the following line, ouput the flight plan you will follow. If there are multiple answers, print any. It is assured that some flight plan exists between Delhi and New York.

---

input

```
6 9 6 4
1 2 2
2 5 10
5 4 5
4 3 2
3 1 1
1 4 1
2 4 3
6 1 3
2 6 5
```

output

```
4
6 1 4
```
