A. Word

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Vasya is very upset that many people on the Net mix uppercase and lowercase letters in one word. That's why he decided to invent an extension for his favorite browser that would change the letters' register in every word so that it either only consisted of lowercase letters or, vice versa, only of uppercase ones. At that as little as possible letters should be changed in the word. For example, the word `HoUse` must be replaced with `house`, and the word `ViP` — with `VIP`. If a word contains an equal number of uppercase and lowercase letters, you should replace all the letters with lowercase ones. For example, `maTRIx` should be replaced by `matrix`. Your task is to use the given method on one given word.

Input

The first line contains a word s — it consists of uppercase and lowercase Latin letters and possesses the length from 1 to 100.

Output

Print the corrected word s. If the given word s has strictly more uppercase letters, make the word written in the uppercase register, otherwise - in the lowercase one.

Sample test(s)

input
HoUse
output
house

input
ViP
output
VIP

input
maTRIx
output
matrix

B. Fortune Telling

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Marina loves Sasha. But she keeps wondering whether Sasha loves her. Of course, the best way to know it is fortune telling. There are many ways of telling fortune, but Marina has picked the easiest one. She takes in her hand one or several camomiles and tears off the petals one by one. After each petal she pronounces alternatively "Loves" and "Doesn't love", at that Marina always starts with "Loves". There are n camomiles growing in the field, possessing the numbers of petals equal to $a_1, a_2, \ldots a_n$. Marina wants to pick a bouquet with the maximal possible total number of petals so that the result would still be "Loves". Help her do that; find the maximal number of petals possible in the bouquet.

Input

The first line contains an integer n ($1 \le n \le 100$), which is the number of flowers growing in the field. The second line contains n integers a_i ($1 \le a_i \le 100$) which represent the number of petals on a given i-th camomile.

Output

Print a single number which is the maximal number of petals in the bouquet, the fortune telling on which would result in "Loves". If there are no such bouquet, print 0 instead. The bouquet may consist of a single flower.

Sample test(s)

input
```
1
1
```
output
```
1
```

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

input
```
3
5 6 7
```
output
```
13
```

C. Title

Vasya has recently finished writing a book. Now he faces the problem of giving it the title. Vasya wants the title to be vague and mysterious for his book to be noticeable among others. That's why the title should be represented by a single word containing at least once each of the first k Latin letters and not containing any other ones. Also, the title should be a palindrome, that is it should be read similarly from the left to the right and from the right to the left.

Vasya has already composed the approximate variant of the title. You are given the title template s consisting of lowercase Latin letters and question marks. Your task is to replace all the question marks by lowercase Latin letters so that the resulting word satisfies the requirements, described above. Each question mark should be replaced by exactly one letter, it is not allowed to delete characters or add new ones to the template. If there are several suitable titles, choose the first in the alphabetical order, for Vasya's book to appear as early as possible in all the catalogues.

Input

The first line contains an integer k ($1 \leq k \leq 26$) which is the number of allowed alphabet letters. The second line contains s which is the given template. In s only the first k lowercase letters of Latin alphabet and question marks can be present, the length of s is from 1 to 100 characters inclusively.

Output

If there is no solution, print `IMPOSSIBLE`. Otherwise, a single line should contain the required title, satisfying the given template. The title should be a palindrome and it can only contain the first k letters of the Latin alphabet. At that, each of those k letters must be present at least once. If there are several suitable titles, print the lexicographically minimal one.

The lexicographical comparison is performed by the standard < operator in modern programming languages. The line a is lexicographically smaller than the line b, if exists such an i ($1 \leq i \leq |s|$), that $a_i < b_i$, and for any j ($1 \leq j < i$) $a_j = b_j$. $|s|$ stands for the length of the given template.

Sample test(s)

input
3 a?c

output
IMPOSSIBLE

input
2 a??a

output
abba

input
2 ?b?a

output
abba

D. Team Arrangement

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Recently personal training sessions have finished in the Berland State University Olympiad Programmer Training Centre. By the results of these training sessions teams are composed for the oncoming team contest season. Each team consists of three people. All the students of the Centre possess numbers from 1 to $3n$, and all the teams possess numbers from 1 to n. The splitting of students into teams is performed in the following manner: while there are people who are not part of a team, a person with the best total score is chosen among them (the captain of a new team), this person chooses for himself two teammates from those who is left according to his list of priorities. The list of every person's priorities is represented as a permutation from **the rest of** $3n$ - 1 students who attend the centre, besides himself.

You are given the results of personal training sessions which are a permutation of numbers from 1 to $3n$, where the i-th number is the number of student who has won the i-th place. No two students share a place. You are also given the arrangement of the already formed teams in the order in which they has been created. Your task is to determine the list of priorities for the student number k. If there are several priority lists, choose the lexicographically minimal one.

Input

The first line contains an integer n ($1 \le n \le 10^5$) which is the number of resulting teams. The second line contains $3n$ space-separated integers from 1 to $3n$ which are the results of personal training sessions. It is guaranteed that every student appears in the results exactly once.

Then follow n lines each containing three integers from 1 to $3n$ — each line describes the members of a given team. The members of one team can be listed in any order, but the teams themselves are listed in the order in which they were created. It is guaranteed that the arrangement is correct, that is that every student is a member of exactly one team and those teams could really be created from the given results using the method described above.

The last line contains number k ($1 \le k \le 3n$) which is the number of a student for who the list of priorities should be found.

Output

Print $3n$ - 1 numbers — the lexicographically smallest list of priorities for the student number k.

The lexicographical comparison is performed by the standard < operator in modern programming languages. The list a is lexicographically less that the list b if exists such an i ($1 \le i \le 3n$), that $a_i < b_i$, and for any j ($1 \le j < i$) $a_j = b_j$. Note, that the list `1 9 10` is lexicographically less than the list `1 10 9`. That is, the comparison of lists is different from the comparison of lines.

Sample test(s)

input
```
3
5 4 1 2 6 3 7 8 9
5 6 2
9 3 4
1 7 8
4
```

output
```
2 3 5 6 9 1 7 8
```

input
```
3
5 4 1 2 6 3 7 8 9
5 6 2
9 3 4
1 7 8
8
```

output
```
1 2 3 4 5 6 7 9
```

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

output
```
5 6 1 2 3
```

E. Shortest Path

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

In Ancient Berland there were n cities and m two-way roads of equal length. The cities are numbered with integers from 1 to n inclusively. According to an ancient superstition, if a traveller visits three cities a_i, b_i, c_i in row, without visiting other cities between them, a great disaster awaits him. Overall there are k such city triplets. Each triplet is ordered, which means that, for example, you are allowed to visit the cities in the following order: a_i, c_i, b_i. Vasya wants to get from the city 1 to the city n and not fulfil the superstition. Find out which minimal number of roads he should take. Also you are required to find one of his possible path routes.

Input

The first line contains three integers n, m, k ($2 \le n \le 3000$, $1 \le m \le 20000$, $0 \le k \le 10^5$) which are the number of cities, the number of roads and the number of the forbidden triplets correspondingly.

Then follow m lines each containing two integers x_i, y_i ($1 \le x_i, y_i \le n$) which are the road descriptions. The road is described by the numbers of the cities it joins. No road joins a city with itself, there cannot be more than one road between a pair of cities.

Then follow k lines each containing three integers a_i, b_i, c_i ($1 \le a_i, b_i, c_i \le n$) which are the forbidden triplets. Each ordered triplet is listed mo more than one time. All three cities in each triplet are distinct.

City n can be unreachable from city 1 by roads.

Output

If there are no path from 1 to n print -1. Otherwise on the first line print the number of roads d along the shortest path from the city 1 to the city n. On the second line print $d + 1$ numbers — any of the possible shortest paths for Vasya. The path should start in the city 1 and end in the city n.

Sample test(s)

input
```
4 4 1
1 2
2 3
3 4
1 3
1 4 3
```
output
```
2
1 3 4
```

input
```
3 1 0
1 2
```
output
```
-1
```

input
```
4 4 2
1 2
2 3
3 4
1 3
1 2 3
1 3 4
```
output
```
4
1 3 2 3 4
```