Same to Same Again

Problem Statement

There are a list of **N** values which were inserted into a **stack** and a list of **M** values which were inserted into a **queue**. After that the elements of the stack and queue are removed and put them into the list where the belong. You need to tell if both of the list are same or not after removing those elements from the stack and queue.

Note: You need to implement Stack and Queue to solve this problem. You can't insert those values to anything else and also you can't use STL here. You can implement stack and queue by linked list or array as you wish.

Input Format

- First line will contain **N** and **M**.
- Second line will contain list **A** with **N** values.
- Third line will contain list **B** with **M** values.

Constraints

- 1 <= **N, M** <= 10^6
- 0 <= Values of list **A** and **B** <= 1000

Output Format

- Output "**YES**" if they were same, otherwise "**NO**".

Sample Input 0

```
5 5

10 20 30 40 50

50 40 30 20 10
```

Sample Output 0

```
YES
```

Sample Input 1

```
4 4

10 20 30 40

10 20 30 40
```

Sample Output 1

```
NO
```

Sample Input 2

```
5 4
```

```
10 20 30 40 50

50 40 30 20
```

Sample Output 2

NO

Same to Same Again and Again

Problem Statement

There are a list of **N** values which were inserted into a **stack** and a list of **M** values which were inserted into a **queue**. After that the elements of the stack and queue are removed and put them into the list where the belong. You need to tell if both of the list are same or not after removing those elements from the stack and queue.

Note: You need to use STL Stack and Queue to solve this problem. You can't insert those values to anything else.

Input Format

- First line will contain **N** and **M**.
- Second line will contain list **A** with **N** values.
- Third line will contain list **B** with **M** values.

Constraints

- 1 <= **N, M** <= 10^6
- 0 <= Values of list **A** and **B** <= 1000

Output Format

- Output "**YES**" if they were same, otherwise "**NO**".

Sample Input 0

```
5 5
10 20 30 40 50
50 40 30 20 10
```

Sample Output 0

```
YES
```

Sample Input 1

```
4 4
10 20 30 40
10 20 30 40
```

Sample Output 1

```
NO
```

Sample Input 2

```
5 4
10 20 30 40 50
50 40 30 20
```

Sample Output 2

```
NO
```

Special Queries

Problem Statement

You will be given **Q** queries. In each query you will get a command. The command is of two types -

1. You will be given **0** and **name** of a person who stood in a line of a ticket counter.
2. You will be given only **1** which means the person in front of the line got the ticket and will be removed from the line. You need to print the name of that person who got that ticket. If there are no one in the line, print "**Invalid**".

Note: There can be multiple person in the line with same name.

Input Format

- First line will contain **Q**.
- Next **Q** lines will contain the commands.

Constraints

1. 1 <= **Q** <= 10^6
2. 1 <= |**name**| <= 1000; Here |name| means the length of the string and it will not contain any space. The string will contain only small English alphabets.

Output Format

- For each time someone get out of the line, print his/her name. Print a new line after that.

Sample Input 0

```
5
0 rahim
0 karim
1
0 sakib
1
```

Sample Output 0

rahim
karim

Sample Input 1

8
1
0 embappe
0 neymar
1
1
0 messi
1
1

Sample Output 1

Invalid
embappe
neymar
messi
Invalid

Sample Input 2

6
0 embappe
0 embappe
1
1
0 messi
1

Sample Output 2

embappe
embappe

messi

Problem Statement

Given a string s containing just the characters '**A**' and '**B**, determine if the input string is valid.

An input string is valid if the string is empty after doing some operatios. The available operations are:

- **B** can delete its previous available character **A** along with itself. If there is no **A** available to delete, it will not delete itself.
- **A** can delete its previous available character **B** along with itself. If there is no **B** available to delete, it will not delete itself.

Input Format

- First line will contain **T**, the number of test cases.
- Next **T** lines will contain the string **S**.

Constraints

1. 1 <= **T** <= 1000
2. 1 <= |**S**| <= 1000; Here |**S**| means the length of the string.

Output Format

- Output "**YES**" if the string is valid, otherwise "**NO**".

Sample Input 0

```
10
AABB
BABA
BBAA
ABAB
AAAB
ABBB
ABBA
BAABAB
BBBAABA
AAABABBABB
```

Sample Output 0

```
YES
```

YES

YES

YES

NO

NO

YES

YES

NO

YES

The Magician and The Magic Colors

Problem Statement

Arif, the magician is putting some magic colors in a box serially. But he is not providing any divider between the colors. So, the color may be mix up **instantly**.

There are three types of colors- **Red, Green and Blue**. How they could mix up is given below -

- Red+Blue = Purple
- Red+Green = Yellow
- Blue+Green = Cyan

And there are some other problems. If two same type of colors mix up, they will vanish each other. For example if two Purple colors get together, both of them will be vanished.

Can you help the magician to get the final colors that will be in the box?

Input Format

- First line will contain **T**, the number of test cases.
- Next line will contain **N**, number of colors in the box.
- Next line will contain **N** characters **(R,G,B only)** , first capital letter of the color.

Constraints

1. 0 < **T** <= 100
2. 0 < **N** <= 100

Output Format

- Output the first capital letter of the colors that are saved finally. If there are no colors in the box, the output could be empty in that case.

Sample Input 0

```
2
3
RBG
4
RGBB
```

Sample Output 0

```
PG
Y
```

Sample Input 1

```
6
6
RGGRRG
6
RGRGRG
4
RGGR
5
RGGGR
4
RGGB
18
RBRBRBGRBRRGBGGBGB
```

Sample Output 1

```
Y
Y

YR
YC

PYPYC
```