Browser History

Problem Statement

You are given a doubly linked list of strings. These strings refer to web **addresses** without any spaces. You will be given Q queries. In each query you will be given some commands. Type of commands are -

1. **visit address** - You need to go to that address from where you are in that list and print that **address** if it is in the list. Otherwise print "**Not Available**".
2. **next** - You need to go to the next address from where you are in that list and print that **address** if it is in the list. Otherwise print "**Not Available**".
3. **prev** - You need to go to the previous address from where you are in that list and print that **address** if it is in the list. Otherwise print "**Not Available**".

Note: You can use **Linked List** or **STL List** to solve this problem.

Input Format

- First line will contain the values of the doubly linked list, and will terminate with the string "**end**".
- Second line will contain **Q**.
- Next Q lines will contain the commands. It is guranteed that you will get "**visit address**" command at first which will contain a valid address. It will not contain valid address everytime!

Constraints

1. 1 <= **N** <= 1000; Here **N** is the maximum number of nodes of the linked list.
2. 1 <= **Q** <= 1000;
3. 1 <= |Address| <= 100; Here |Address| is the length of the string address.

Output Format

- For each query output as asked.

Sample Input 0

```
facebook google phitron youtube twitter end
12
visit phitron
prev
prev
prev
prev
next
visit twitter
next
next
prev
visit django
Prev
```

Sample Output 0

```
phitron
google
facebook
Not Available
Not Available
google
twitter
Not Available
Not Available
youtube
Not Available

phitron
```

Queries Again

Problem Statement

You have a doubly linked list which is **empty** initially. Then you will be given **Q** queries. In each query you will be given two values **X** and **V**.

- You need to insert the value **V** at index **X**. Assume that index starts from 0.

- After that for each query you need to print the linked list from left to right and right to left.
- If the index is invalid, then print "Invalid".

Note: You must use **doubly linked list**, otherwise you will not get marks.

Input Format

- First line will contain **Q**.
- Next **Q** lines will contain **X** and **V**.

Constraints

1. 1 <= **Q** <= 1000;
2. 0 <= **X** <= 1000;
3. 0 <= **V** <= 1000

Output Format

- For each query print the linked list from left to right and right to left or print "Invalid" as asked.
- Print "**L ->** " before printing the linked list from left to right.
- Print "**R ->** " before printing the linked list from right to left.

Sample Input 0

```
5
1 10
0 10
1 20
3 30
2 30
```

Sample Output 0

```
Invalid
L -> 10
R -> 10
L -> 10 20
R -> 20 10
Invalid
L -> 10 20 30
R -> 30 20 10
```

Sample Input 1

```
10
0 10
1 20
```

```
0 30

1 40

6 50

0 60

4 70

4 80

2 90

1 100
```

Sample Output 1

```
L -> 10

R -> 10

L -> 10 20

R -> 20 10

L -> 30 10 20

R -> 20 10 30

L -> 30 40 10 20

R -> 20 10 40 30

Invalid

L -> 60 30 40 10 20

R -> 20 10 40 30 60
```

```
L -> 60 30 40 10 70 20

R -> 20 70 10 40 30 60

L -> 60 30 40 10 80 70 20

R -> 20 70 80 10 40 30 60

L -> 60 30 90 40 10 80 70 20

R -> 20 70 80 10 40 90 30 60

L -> 60 100 30 90 40 10 80 70 20

R -> 20 70 80 10 40 90 30 100 60
```

Palindrome

Problem Statement

You need to take a singly linked list of integer value as input. You need to tell if the singly linked list forms a palindrome or not.

Note: You need to solve this using **singly linked list**, otherwise you will not get marks.

Input Format

- Input will contain the values of the singly linked list, and will terminate with -1.

Constraints

1. 1 <= **N** <= 10^6; Here N is the maximum number of nodes of the linked list.

2. 0 <= **V** <= 1000; Here V is the value of each node.

Output Format

- Output "**YES**" if it forms a palindrom otherwise print "**NO**".

Sample Input 0

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

Sample Output 0

```
YES
```

Sample Input 1

```
1 2 2 1 -1
```

Sample Output 1

```
YES
```

Sample Input 2

```
1 -1
```

Sample Output 2

```
YES
```

Sample Input 3

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

Sample Output 3

```
NO
```

Remove Duplicates II

Problem Statement

You need to take a linked list of integer value as input using STL List. You need to remove the duplicate values and print the unique values in ascending order.

Note: You need to solve this using **STL list**, otherwise you will not get marks.

Input Format

- Input will contain the values of the linked list, and will terminate with -1.

Constraints

1. 1 <= **N** <= 10^5; Here N is the maximum number of nodes of the linked list.
2. 0 <= **V** <= 1000; Here V is the value of each node.

Output Format

- Output the new linked list in ascending order.

Sample Input 0

1 2 1 3 2 4 2 1 3 -1

Sample Output 0

1 2 3 4

Sample Input 1

2 2 2 2 -1

Sample Output 1

2

Sample Input 2

10 4 2 1 5 6 -1

Sample Output 2

1 2 4 5 6 10

Sample Input 3

5 4 3 5 4 2 5 3 2 4 5 1 2 3 4 5 -1

Sample Output 3

1 2 3 4 5

Sample Input 4

4 3 5 4 2 3 2 4 1 2 3 4 -1

Sample Output 4

```
1 2 3 4 5
```

Remove Duplicates I

Problem Statement

You need to take a singly linked list of integer value as input. You need to remove the duplicate values and print the unique values in ascending order.

Note: You need to solve this using **singly linked list**, otherwise you will not get marks.

Input Format

- Input will contain the values of the singly linked list, and will terminate with -1.

Constraints

1. 1 <= **N** <= 10^3; Here N is the maximum number of nodes of the linked list.
2. 0 <= **V** <= 1000; Here V is the value of each node.

Output Format

- Output the new singly linked list in ascending order.

Sample Input 0

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

Sample Output 0

```
1 2 3 4
```

Sample Input 1

```
2 2 2 2 -1
```

Sample Output 1

```
2
```

Sample Input 2

```
10 4 2 1 5 6 -1
```

Sample Output 2

```
1 2 4 5 6 10
```

Sample Input 3

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

Sample Output 3

```
1 2 3 4 5
```

Sample Input 4

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

Sample Output 4

```
1 2 3 4 5
```