Matrix

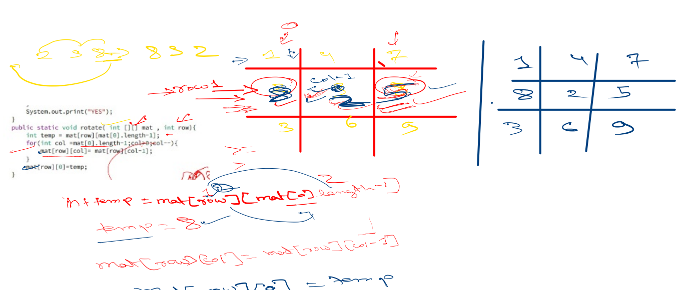
String

String   Content

~~Cla~~
travers
reverse
substring

```
str = "sagar"
rev = "";
① for(int i = str.length()-1; i>=0; i--)
{
    syso(        )
    rev += str.charAt(i);
}
```

rev



Given two strings str & target, return the index where target string occurs for the first time in String str.

Input Format

- The first line contains the string str.
- The second line contains the string target.

Constraints

- 1 <= str.length <= 10^4
- str consists of lowercase English letters.

Output Format

- Print the index where the target string occurs for the first time.
- If the target string is not found, print -1.

Sample Input 0

```
geekster
st
```

Sample Output 0

```
4
```

Explanation 0

The string "geekster" contains the target string "st" from index [4-5]. So, the starting index (4) is printed as the result.

Sample Input 1

```
geekster
ab
```

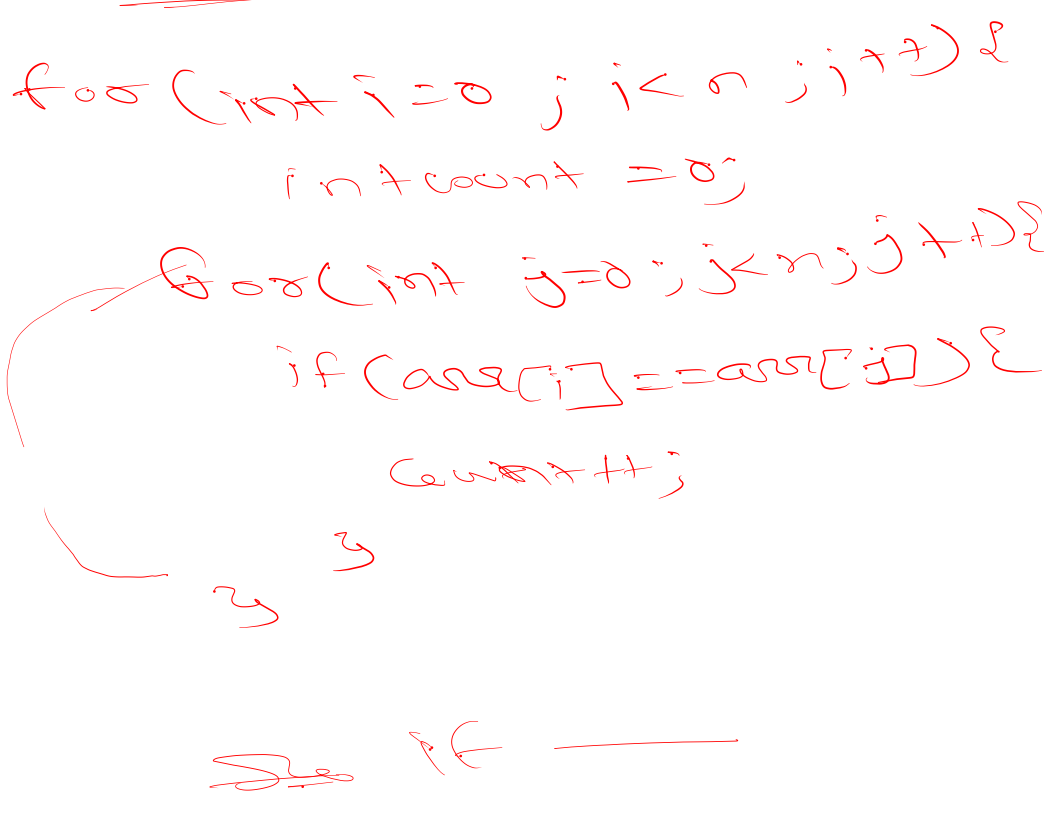
Sample Output 1

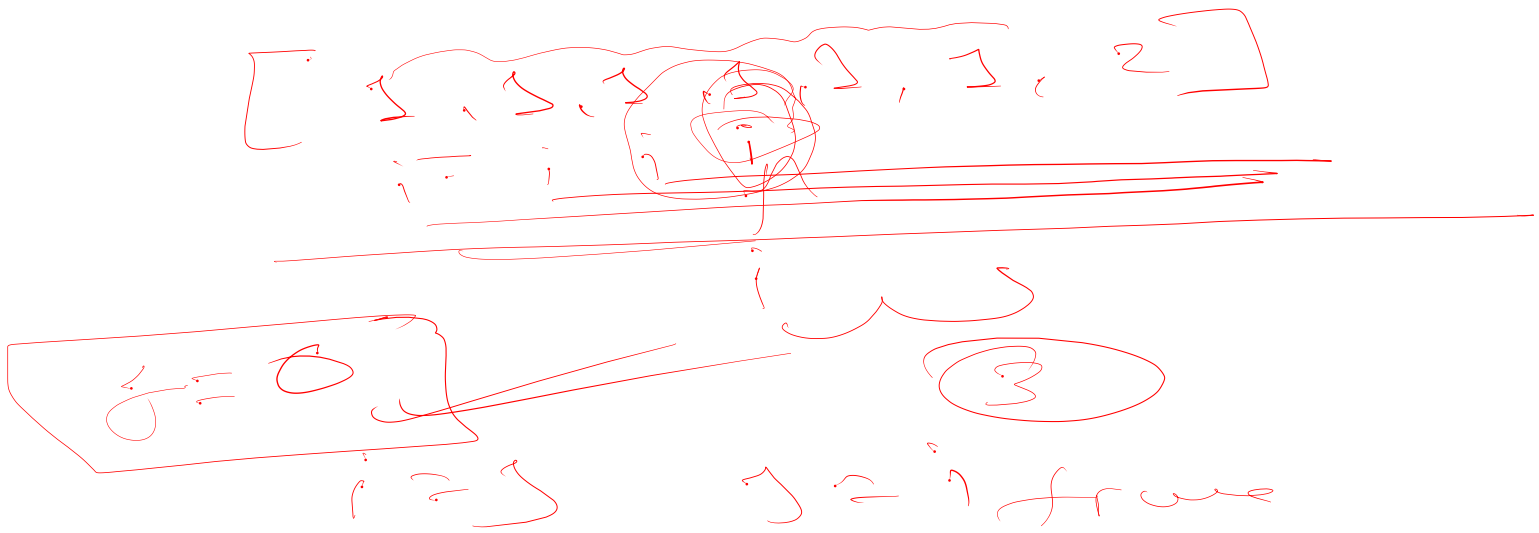
```
-1
```

Explanation 1

String "geekster" does not contain "ab". So, result is -1.

```
public static boolean
threeTimesOccur(int[] arr, int n) {

    for (int i = 0; i < n; i++) {
      int count = 0;


      for (int j = 0; j < n; j++) {
        if (arr[i] == arr[j]) {
          count++;
        }
      }

      if (count == 3) {
        return true;
      }
    }

    return false;
  }
```





Given an array of integers **arr**, replace each element with its rank.

The rank represents how large the element is. The rank has the following rules:

- Rank is an integer starting from 1.
- The larger the element, the larger the rank. If two elements are equal, their rank must be the same.
- Rank should be as small as possible.

Input Format

First line contains an integer n.

Second line contains an array of integers of size n.

Constraints

0 <= arr.length <= 10^5

109 <= arr[i] <= 10^9

Output Format

Returns an array.

Sample Input 0

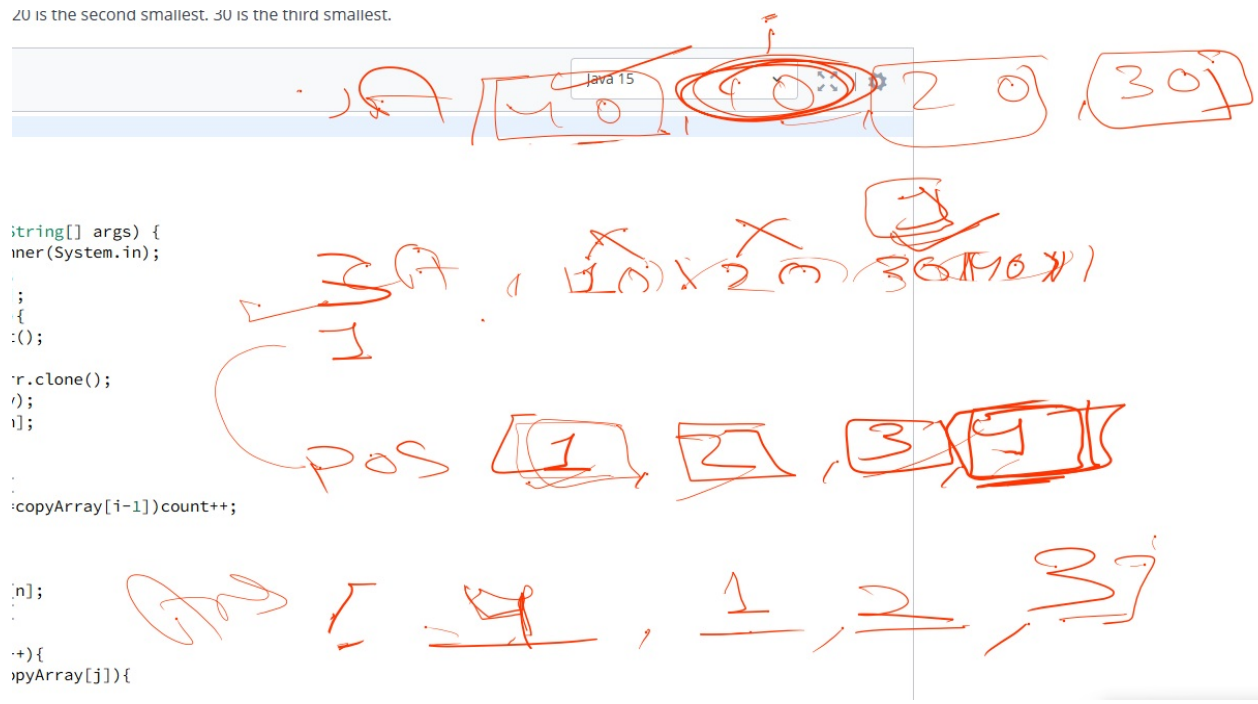
```
4
40 10 20 30
```

Sample Output 0

```
4 1 2 3
```

Explanation 0

40 is the largest element. 10 is the smallest. 20 is the second smallest. 30 is the third smallest.




```java
import java.io.*;
import java.util.*;

public class Solution {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int arr[]= new int[n];
        for(int i =0;i<n;i++){
            arr[i]=sc.nextInt();
        }
        int copyArray [] = arr.clone();
        Arrays.sort(copyArray);
        int pos[] = new int[n];
        int count =1;
        pos[0]=count;
        for(int i=1;i<n;i++){
            if(copyArray[i]!=copyArray[i-1])count++;
            pos[i]=count;
        }

        int ans [] = new int[n];
        for(int i=0;i<n;i++){
            int idx =0;
           for(int j=0;j<n;j++){
               if(arr[i]== copyArray[j]){
                   idx=j;
                   break;
               }
           }
            ans[i]= pos[idx];
        }

         for(int i =0;i<n;i++){

             System.out.print(ans[i]+" ");                          }
    }
}
```

A palindrome is a term that can be read the s
punctuation or capitalization changes. To dete
converted to lowercase, and all non-alphanun

Input Format

string str as an input.

Constraints

1 <= str.length <= 2 * 10^5

str consists only of printable ASCII characters.

Output Format

return true or false.

Sample Input 0

```
A man, a plan, a canal: Panama
```

Sample Output 0

```
true
```

```java
import java.io.*;
import java.util.*;

public class Solution {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String str = sc.nextLine();

        str = str.toLowerCase().replaceAll("[^a-z0-9]","");
        int i =0;
        int j = str.length()-1;
        while(i<=j){
            if(str.charAt(i)!=str.charAt(j)){
                System.out.print(false);
                return;
            }
            i++;
            j--;
        }

        System.out.print(true);
    }
}
```

amanaplanacanalpanama


```java
import java.io.*;
import java.util.*;

public class Solution {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String str = sc.nextLine();

        str = str.toLowerCase().replaceAll("[^a-z0-9]","");

        int i =0;
        int j = str.length()-1;
        while(i<=j){
            if(str.charAt(i)!=str.charAt(j)){
                System.out.print(false);
                return;
            }
            i++;
            j--;
        }

        System.out.print(true);
    }
}
```