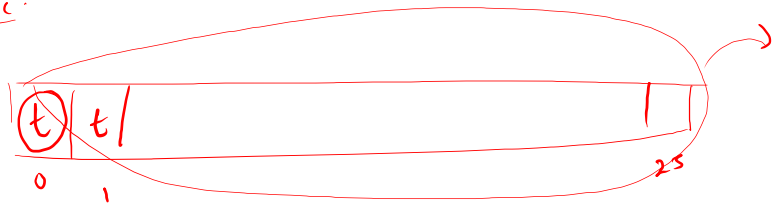
a, b, c --- z

A B C --- Z

"the quick brown fox jumps over the lazy dog"

pangram.

Logic:



boolean

c = 0;

freq[i] == true

c++;

if (c == 26)

{

}

Check Anagram

Problem | Submissions | Leaderboard | Discussions

Take two **Strings** as input and check whether they are anagram or not. Print **True** if they are anagram else print **False**.

Anagram: An anagram is a word or phrase formed by rearranging the letters of a different word or phrase, typically using all the original letters exactly once.

Eg: **"peek"** and **"keep"** are anagrams.

Note: All characters will be in lowercase.

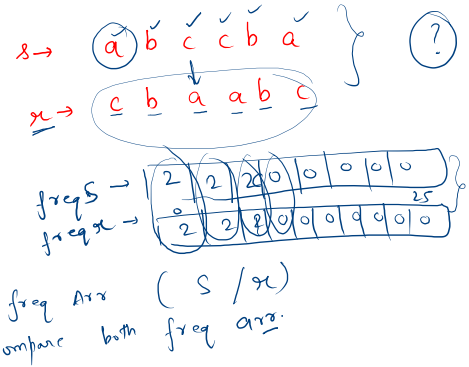
Sample Input 0

```
abccba
cbaabc
```

Sample Output 0

```
True
```

Explanation 0




```java
import java.io.*;
import java.util.*;
import java.text.*;
import java.math.*;
import java.util.regex.*;

public class Solution {
    public static boolean isAnagram(int [] freqS , int [] freqR){
        for(int i = 0; i < 26; i++){
            if(freqS[i] != freqR[i])
                return false;
        }
        return true;
    }

    public static void main(String[] args) {
        Scanner scn = new Scanner(System.in);
        String s = scn.next();
        String r = scn.next();

        int [] freqArrS = new int[26];
        int [] freqArrR = new int[26];

        for(int i = 0; i < s.length(); i++){
         char ch = s.charAt(i);
            int idx = ch - 'a';
            freqArrS[idx]++;
        }
```

```java
    for(int i = 0; i < r.length(); i++){
     char ch = r.charAt(i);
        int idx = ch - 'a';
        freqArrR[idx]++;
    }
    boolean ans = isAnagram(freqArrS, freqArrR);
    if(ans){
        System.out.println("True");
    }
    else{
        System.out.println("False");
    }
```

Isogramic String

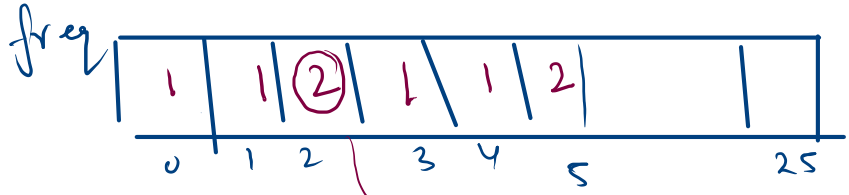
Problem | Submissions | Leaderboard | Discussions

Given a string s in its lower-case, you have to decide if its an Isogram or not. A string is said to be an Isogram, if all the letters in the string, occur only once in it.

Sample Input 0

```
geekster
```

eg. "abccdeff"



1. creat freq arr.
2. check if (freq[i] > 1)
   → return. false.

K Frequent Characters

Problem | Submissions | Leaderboard | Discussions

Given a string and an integer k, print the first k high frequency characters.

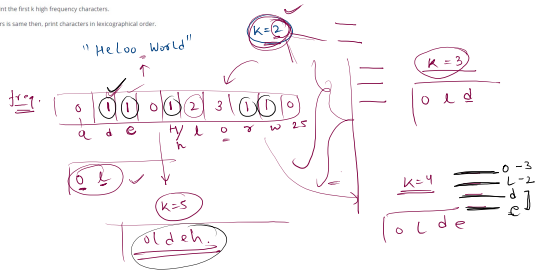
Note: if frequency of two characters is same then, print characters in lexicographical order.

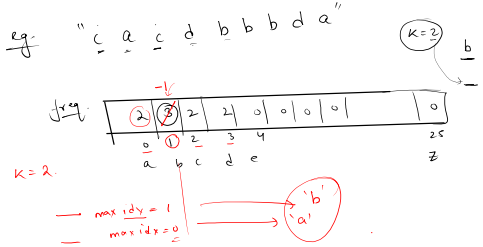
Sample Input 0

```
Heloo World
2
```

Sample Output 0

```
o l
```






```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    String s = scn.nextLine();
    int k = scn.nextInt();

    s = s.toLowerCase();

    int [] freq = new int[26];
    for(int i = 0; i < s.length(); i++){
        char ch = s.charAt(i);
        if(ch != ' '){

            int idx = ch - 'a';
            freq[idx]++;
        }
    }

    for(int i = 1; i <= k; i++){

        int max = 0;
        int maxIdx = -1;

        for(int p = 0; p < 26; p++){
            if (freq[p] > max){
                max = freq[p];
                maxIdx = p;
            }

        }

        System.out.print((char)(maxIdx + 'a') + " ");
        freq[maxIdx] = -1;
```

First Non-Repeating Character

Problem | Submissions | Leaderboard | Discussions

Find first non-repeating character of string.

Eg: AAccddeefgg. So the answer here is f

Note: If first non-repeating character of string is not found print -1.

Sample Input 0

```
AAccddeefgg
```

Sample Output 0

```
f
```

eg. aabb ?

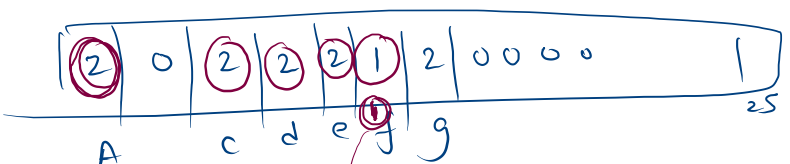
-1

eg abcd

?

'a'

AAccddeefgg → lower case.



=

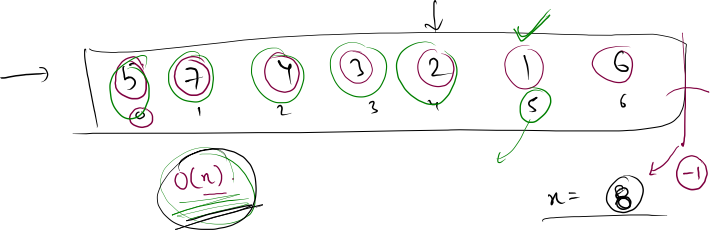
f

if freq[i]==1
return i

-1

ccAAddfee

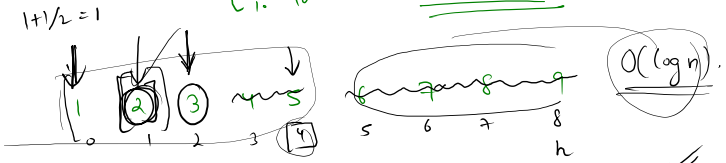
f



Binary Search

↳ compulsory

↳ i.e. to have sorted arr.



0+1/2 =

0+1/2 = 0

0+4/2 = (2)

mid = 4

l+h/2 = 8+0/2 = 4

x = 2

arr[4] = (5)

A → 2 > arr[4]

B → 2 < arr[4]