

题目描述：

给定两个字符串 `str1` 和 `str2`，如果字符串 `str1` 中的字符，经过排列组合后的字符串中，只要有一个字符串是 `str2` 的子串，则认为 `str1` 是 `str2` 的关联子串。

若 `str1` 是 `str2` 的关联子串，请返回子串在 `str2` 的起始位置；

若不是关联子串，则返回-1。

示例 1：

输入：`str1="abc",str2="efghicabiiii"`

输出：5

解释：`str2` 包含 `str1` 的一种排列组合（"cab"），此组合在 `str2` 的字符串起始位置为 5（从 0 开始计数）

示例 2：`str1="abc",str2="efghicaibii"`

输出：-1。

预制条件：

输入的字符串只包含小写字母；

两个字符串的长度范围[1, 100,000]之间

若 `str2` 中有多个 `str1` 的组合子串，请返回第一个子串的起始位置。

输入描述：

输入两个字符串，分别为题目中描述的 `str1`、`str2`。

输出描述：

如果 `str1` 是 `str2` 的关联子串，则返回子串在 `str2` 中的起始位置。

如果 `str1` 不是 `str2` 的关联子串，则返回-1。

若 `str2` 中有多个 `str1` 的组合子串，请返回最小的起始位置。

补充说明：

输入的字符串只包含小写字母；

两个字符串的长度范围[1, 100,000]之间

示例 1

输入：

`abc efghicabiiii`

输出：

5

说明：

`str2` 包含 `str1` 的一种排列组合（"cab"），此组合在 `str2` 的字符串起始位置为 5（从 0 开始计数）

示例 2

输入：

`abc efghicaibii`

输出：

-1

说明：

“abc”字符串中三个字母的各种组合（abc、acb、bac、bca、cab、cba），`str2` 中均不包含，因此返回-1

```

import java.util.*;

public class Main {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        String text = in.nextLine();
        String str1 = text.split(" ")[0];
        String str2 = text.split(" ")[1];

        LinkedList<String> list = new LinkedList<String>(Arrays.asList(str1.split("")));

        boolean flag = false;

        int index = 0;

        for (int i = 0; i < str2.length(); i++) {
            if (list.contains(String.valueOf(str2.charAt(i))) && !flag) {
                list = new LinkedList<String>(Arrays.asList(str1.split("")));
                flag = true;
                index = i;
            }

            if (flag) {
                if (list.contains(String.valueOf(str2.charAt(i)))) {
                    list.remove(String.valueOf(str2.charAt(i)));
                    if (list.size() == 0) {
                        System.out.println(index);
                        return;
                    }
                } else {
                    flag = false;
                }
            }
        }

        System.out.println(-1);
    }
}

```