

JAVA-字符匹配

题目描述：

给你一个字符串数组（每个字符串均由小写字母组成）和一个字符规律（由小写字母和.和*组成），识别数组中哪些字符串可以匹配到字符规律上。

'.' 匹配任意单个字符，'*' 匹配零个或多个任意字符；判断字符串是否匹配，是要涵盖整个字符串的，而不是部分字符串。

输入描述：

第一行为空格分割的多个字符串，1<单个字符串长度<100，1<字符串个数<100

第二行为字符规律，1<=字符规律长度<=50

不需要考虑异常场景

输出描述：

匹配的字符串在数组中的下标（从0开始），多个匹配时下标升序并用,分割，若均不匹配输出-1

补充说明：

示例 1

输入：

ab aab abacd

.*

输出：

0,1,2

说明：

ab 中 a 匹配. b 匹配* 可以全匹配; aab 中 a 匹配. ab 匹配* 可以全匹配; abacd 中 a 匹配. bacd 匹配* 可以全匹配；输出对应字符串数组下标 0,1,2

示例 2

输入：

ab aab

a.b

输出：

1

说明：

aab 中第一个 a 匹配 a 第二个 a 匹配. b 匹配 b 可以全匹配； 输出对应字符串数组下标 1

```
import java.util.Scanner;
```

```
// 注意类名必须为 Main, 不要有任何 package xxx 信息
```

```
public class Main {
```

```
    public static void main(String[] args) {
```

```
//        boolean ab = isMatch("aab", 0, ".*", 0);
```

```
//        System.out.println(ab);
```

```
        ques3(args);
```

```
    }
```

```
    public static void ques3(String[] args) {
```

```
        Scanner in = new Scanner(System.in);
```

```
// 注意 hasNext 和 hasNextLine 的区别
```

```
        String s1 = in.nextLine();
```

```
        String[] matchStrArr = s1.split(" ");
```

```
        String s2 = in.nextLine();
```

```
        StringBuilder sb = new StringBuilder();
```

```
        for (int i = 0; i < matchStrArr.length; i++) {
```

```
            boolean match = isMatch(matchStrArr[i], 0, s2, 0);
```

```
            if(match){
```

```
                sb.append(i).append(",");
```

```
            }
```

```
        }
```

```
        if(sb.length() > 0){
```

```
            sb.deleteCharAt(sb.length() - 1);
```

```
            System.out.println(sb.toString());
```

```
        } else {
```

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

```
        }
```

```
    }
```

```
    public static boolean isMatch(String s1, int startIndex1, String s2, int startIndex2){
```

```
        if(startIndex1 == s1.length()){
```

```
            if(s2.length() == startIndex2){
```

```
                return true;
```

```
            }
```

```
            String substring = s2.substring(startIndex2);
```

```
            for(int i = 0; i < substring.length(); i++){
```

```
                if(substring.charAt(i) != '*'){
```

```

        return false;
    }
}

return true;
}

if(startIndex2 == s2.length()){
    if(s1.length() == startIndex1){
        return true;
    }

    return false;
}

if(s2.charAt(startIndex2) == '.'){
    // 匹配一个字符
    boolean match = isMatch(s1, startIndex1 + 1, s2, startIndex2 + 1);
    return match;
}

if(s2.charAt(startIndex2) == '*'){
    for(int matchSize = 0; matchSize <= s1.length() - startIndex1; matchSize++){
        boolean match = isMatch(s1, startIndex1 + matchSize, s2, startIndex2 + 1);
        if(!match){
            continue;
        }
        return match;
    }
}

if(s1.charAt(startIndex1) != s2.charAt(startIndex2)){
    return false;
}

boolean match = isMatch(s1, startIndex1 + 1, s2, startIndex2 + 1);
return match;
}
}

```