Java-字符串位运算-给定一个元素类型为小写字符串的数组
题目描述：
给定一个元素类型为小写字符串的数组，请计算两个没有相同字符的元素 长度乘积的最大值，如果没有符合条件的两个元素，返回 0。
输入描述：
输入为一个半角逗号分隔的小写字符串的数组，2 <= 数组长度<=100，0 < 字符串长度<= 50。
输出描述：
两个没有相同字符的元素 长度乘积的最大值。
补充说明：
示例 1
输入：
iwdvpbn,hk,iuop,iikd,kadgpf
输出：
14
说明：
数组中有 5 个元素。
iwdvpbn 与 hk 无相同的字符，满足条件，iwdvpbn 的长度为 7，hk 的长度为 2，乘积为 14（7*2）。
iwdvpbn 与 iuop、iikd、kadgpf 均有相同的字符，不满足条件。
iuop 与 iikd、kadgpf 均有相同的字符，不满足条件。
iikd 与 kadgpf 有相同的字符，不满足条件。
因此，输出为 14。

```
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) {
        Scanner in = new Scanner(System.in);
        while (in.hasNextLine()) {
            String str = in.nextLine();
            String[] split = str.split(",");
            boolean[][] arr = new boolean[split.length][26];
            int max = 0;
            for (int i = 0; i < split.length; i++) {
                String ss = split[i];
                for (Character c : ss.toCharArray()) {
                    arr[i][c - 'a'] = true;
                }
            }
            for (int i = 0; i < split.length - 1; i++) {
                for (int j = i + 1; j < split.length; j++) {
                    if (!checkHasSameChar(arr, i, j)) {
                        max = Math.max(max, split[i].length() * split[j].length());
                    }
```

```
                    }
                }
                System.out.println(max);
            }
        }

        public static boolean checkHasSameChar(boolean[][] arr, int i, int j) {
            for (int index = 0; index < 26; index++) {
                if (arr[i][index] && arr[j][index]) {
                    return true;
                }
            }
            return false;
        }
}
```