

题目描述：

给定一个整型数组，请从该数组中选择 **3** 个元素组成最小数字并输出（如果数组长度小于 **3**，则选择数组中所有元素来组成最小数字）。

输入描述：

一行用半角逗号分割的字符串记录的整型数组， $0 < \text{数组长度} \leq 100$ ， $0 < \text{整数的取值范围} \leq 10000$ 。

输出描述：

由 **3** 个元素组成的最小数字，如果数组长度小于 **3**，则选择数组中所有元素来组成最小数字。

补充说明

示例1

输入：21,30,62,5,31

输出：21305

说明：数组长度超过3，需要选3个元素组成最小数字，21305由21,30,5三个元素组成的数字，为所有组合中最小的数字

示例2

输入：5,21

输出：215

说明：数组长度小于3，选择所有元素来组成最小值，215为最小值。

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

```
public class Main {
```

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

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

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

```
        String[] strings = s.split(",");
```

```
        int N = strings.length;
```

```
        if (N == 0) {
```

```
            return;
```

```
        }
```

```
        if (N == 1) {
```

```
            System.out.print(s);
```

```

        return;
    }
    if (N == 2) {
        Arrays.sort(strings);
        System.out.print(strings[0] + strings[1]);
        return;
    }

    int M = 6;
    ArrayList[] stringList = new ArrayList[M];
    for (int i = 0; i < M; i++) {
        stringList[i] = new ArrayList();
    }
    for (String string : strings) {
        stringList[string.length() - 1].add(string);
    }

    // 取值方式
    // 1.优先取数位少的数字
    // 2.取字典排序低的数字

    String[] ans = new String[3];
    int loop = 0;
    int n = 0;
    while (loop < 3 && n < M) {
        ans[loop] = getOne(stringList[n]);
        if (ans[loop] == null) {
            n++;
        } else {
            loop++;
        }
    }
    Arrays.sort(ans);
    System.out.print(ans[0] + ans[1] + ans[2]);
}

private static String getOne(ArrayList<String> strings) {
    if (strings.isEmpty()) {
        return null;
    } else {
        Object[] objects = strings.toArray();
        Arrays.sort(objects);
        String res = objects[0].toString();
    }
}

```

```
int i = 0;
while (i < strings.size()) {
    if (Objects.equals(strings.get(i), res)) {
        strings.remove(i);
        return res;
    } else {
        i++;
    }
}
return res;
}
}
```