

题目描述：

给定一组整数（非负），重排顺序后输出一个最大的整数。

示例 1:

输入: [10,9]

输出: 910

说明: 输出结果可能非常大，所以你需要返回一个字符串而不是整数。

输入描述：

数字组合

输出描述：

最大的整数

补充说明：

示例1

输入: 10 9

输出: 910

说明:

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

```
public class Main {
```

```
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        String s = in.nextLine();
        String[] s1 = s.split(" ");
        Arrays.sort(s1, new Comparator<String>() {
            @Override
            public int compare(String o1, String o2) {
                // 优化的代码 这样比较有问题
                // 特殊的字典序
                // int cmp = Math.min(o1.length(),o2.length());
                // int i = 0;
                // 字典序排序
                // while (i<cmp){
                //     if(o1.charAt(i)==o2.charAt(i)){
                //         i++;
                //     }else{
```

```

//          return o1.charAt(i)>o2.charAt(i)?-1:1;
//      }
//  }
//  // 说明一边是另一数的前缀 看较长的一方
//  //return o1.length()<=o2.length()?-1:1;
//  if(o1.length()>o2.length()){
//      int j=0;
//      while (i<o1.length()){
//          if(o1.charAt(i)==o1.charAt(j)){
//              i++;j++;
//          }else{
//              return o1.charAt(i)>o1.charAt(j)?-1:1;
//          }
//      }
//  }else{
//      int j=0;
//      while (i<o2.length()){
//          if(o2.charAt(i)==o2.charAt(j)){
//              i++;j++;
//          }else{
//              return o2.charAt(i)>o2.charAt(j)?1:-1;
//          }
//      }
//  }

String s1 = o1 + o2;
String s2 = o2 + o1;
int k = 0;
while (k < s1.length()) {
    if (s1.charAt(k) == s2.charAt(k)) {
        k++;
    } else {
        return s1.charAt(k) > s2.charAt(k) ? -1 : 1;
    }
}
return 1;
}

});
String ans = "";
for (int i = 0; i < s1.length; i++) {
    ans += s1[i];
}
// 第一位是 0 说明只有 0 直接输出 0
if(ans.charAt(0) == '0') {

```

```
        ans = "0";  
    }  
    System.out.println(ans);  
}  
  
}
```