26012 字符集合

链接：https://www.nowcoder.com/questionTerminal/784efd40ed8e465a84821c8f3970b7b5

【题目解析】：

本题描述很简单，题目描述很清楚，读题即可

【解题思路】：

使用 StringBuilder 构造字符串即可.

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

    Scanner scanner = new Scanner(System.in);
    while (scanner.hasNext()) {
        String str = scanner.nextLine();
        StringBuilder builder = new StringBuilder();
        for (int i = 0; i < str.length(); i++) {
            char temp = str.charAt(i);
            if (!builder.toString().contains(temp + "")) {
                builder.append(temp);
            }
        }

        System.out.println(builder.toString());
    }
}
```

23267-合并两个排序的链表

https://www.nowcoder.com/questionTerminal/d8b6b4358f774294a89de2a6ac4d9337

【题目解析】：

课件上重点讲解的问题.

【解题思路】：

```
public class Solution {
    public ListNode Merge(ListNode list1,ListNode list2) {
        if(list1 == null){
            return list2;
        }
        if(list2 == null){
            return list1;
        }
        ListNode mergeHead = null;
        ListNode current = null;
        while(list1!=null && list2!=null){
            if(list1.val <= list2.val){
                if(mergeHead == null){
                    mergeHead = current = list1;
                }else{
                    current.next = list1;
                    current = current.next;
```

```
                }
                list1 = list1.next;
            }else{
                if(mergeHead == null){
                    mergeHead = current = list2;
                }else{
                    current.next = list2;
                    current = current.next;
                }
                list2 = list2.next;
            }
        }
        if(list1 == null){
            current.next = list2;
        }else{
            current.next = list1;
        }
        return mergeHead;
    }
}
```