**index.wxss**

.container {  
  padding: 20px;  
}  
  
.section {  
  margin-bottom: 20px;  
}  
  
textarea {  
  width: 100%;  
  height: 100px;  
  border: 1px solid #ccc;  
  padding: 10px;  
  box-sizing: border-box;  
}  
  
input {  
  width: 100%;  
  height: 40px;  
  border: 1px solid #ccc;  
  padding: 5px;  
  box-sizing: border-box;  
}  
  
button {  
  width: 100%;  
  height: 40px;  
  background-color: #1aad19;  
  color: white;  
  border: none;  
  text-align: center;  
  line-height: 40px;  
  font-size: 16px;  
  margin-bottom: 10px;  
}

**index.js**

Page({  
  data: {  
    names: '',  
    leaders: '',  
    fixedGroup: '',  
    groupCount: 0,  
    groupedNames: []  
  },  
  
  inputNames(e) {  
    this.setData({  
      names: e.detail.value  
    });  
  },  
  
  inputLeaders(e) {  
    this.setData({  
      leaders: e.detail.value  
    });  
  },  
  
  inputFixedGroup(e) {  
    this.setData({  
      fixedGroup: e.detail.value  
    });  
  },  
  
  inputGroupCount(e) {  
    let value = parseInt(e.detail.value);  
    this.setData({  
      groupCount: isNaN(value) || value < 1 ? 1 : value  
    });  
  },  
  
  groupStudents() {  
    let namesArray = this.data.names.split(",").map(name => name.trim());  
    let leadersArray = this.data.leaders.split(",").map(leader => leader.trim());  
    let fixedGroupArray = this.data.fixedGroup.split(",").map(member => member.trim());  
    let groupCount = this.data.groupCount;  
  
    if (groupCount > namesArray.length) {  
      groupCount = namesArray.length;  
      this.setData({ groupCount: groupCount });  
    }  
  
    let groupedNames = [];  
  
    // 初始化分组数组  
    for (let i = 0; i < groupCount; i++) {  
      groupedNames.push([]);  
    }  
  
    // 确保固定组的成员在同一组，将其放在第一个可用的组  
    if (fixedGroupArray.length > 0) {  
      groupedNames[0] = groupedNames[0].concat(fixedGroupArray);  
      namesArray = namesArray.filter(name => !fixedGroupArray.includes(name));  
      leadersArray = leadersArray.filter(leader => !fixedGroupArray.includes(leader)); // 如果组长在固定组成员中，移除以防重复分配  
    }  
  
    // 打乱剩余的学生名单  
    namesArray = namesArray.sort(() => Math.random() - 0.5);  
  
    // 分配组长到各组，每组只能有一个组长  
    leadersArray.forEach((leader, index) => {  
      let targetGroupIndex = groupedNames.findIndex(group => group.length === 0);  
      if (targetGroupIndex === -1) {  
        targetGroupIndex = index % groupCount;  
      }  
      groupedNames[targetGroupIndex].push(leader);  
      namesArray = namesArray.filter(name => name !== leader); // 剔除已分配的组长  
    });  
  
    // 分配剩余学生到各组，保持平衡  
    let groupIndex = 0;  
    while (namesArray.length > 0) {  
      // 找到当前人数最少的组  
      let minGroupSize = Math.min(...groupedNames.map(group => group.length));  
      let targetGroup = groupedNames.find(group => group.length === minGroupSize);  
      targetGroup.push(namesArray.shift());  
      groupIndex++;  
    }  
  
    // 更新分组结果  
    this.setData({  
      groupedNames: groupedNames  
    });  
  },  
  
  resetGroups() {  
    this.setData({  
      groupedNames: [],  
      names: '',  
      leaders: '',  
      fixedGroup: '',  
      groupCount: 0  
    });  
  },  
  
  clearInputs() {  
    this.setData({  
      names: '',  
      leaders: '',  
      fixedGroup: '',  
      groupCount: 0  
    });  
  }  
});

**index.wxml**

<view class="container">  
  <view class="section">  
    <text>请输入学生姓名（用逗号分隔）:</text>  
    <textarea placeholder="请输入学生姓名" value="{{names}}" bindinput="inputNames"></textarea>  
  </view>  
  
  <view class="section">  
    <text>请输入组长姓名（用逗号分隔）:</text>  
    <textarea placeholder="请输入组长姓名" value="{{leaders}}" bindinput="inputLeaders"></textarea>  
  </view>  
    
  <view class="section">  
    <text>请输入需要在同一组的成员（用逗号分隔）:</text>  
    <textarea placeholder="请输入成员姓名" value="{{fixedGroup}}" bindinput="inputFixedGroup"></textarea>  
  </view>  
    
  <view class="section">  
    <text>请输入分组总数:</text>  
    <input type="number" value="{{groupCount}}" placeholder="请输入分组数" bindinput="inputGroupCount"/>  
  </view>  
  
  <view class="section">  
    <button bindtap="groupStudents">生成分组</button>  
    <button bindtap="resetGroups">重置</button>  
    <button bindtap="clearInputs">清空输入框</button>  
  </view>  
    
  <view class="section">  
    <text>分组结果:</text>  
    <view wx:for="{{groupedNames}}" wx:for-item="group" wx:key="index">  
      <view>组 {{index + 1}}:</view>  
      <view wx:for="{{group}}" wx:for-item="name" wx:key="name">  
        <text>{{name}}</text>  
      </view>  
      <view style="margin-bottom: 10px;"></view>  
    </view>  
  </view>  
</view>