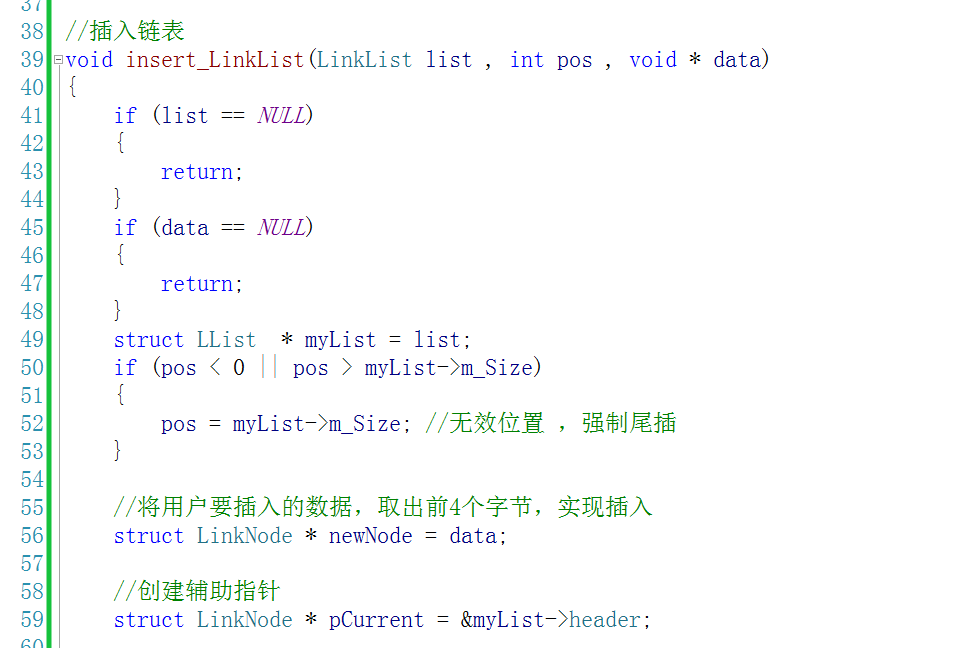
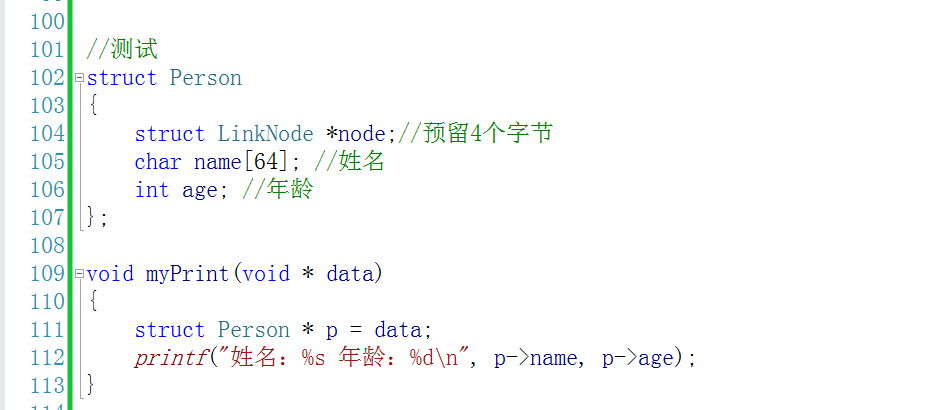
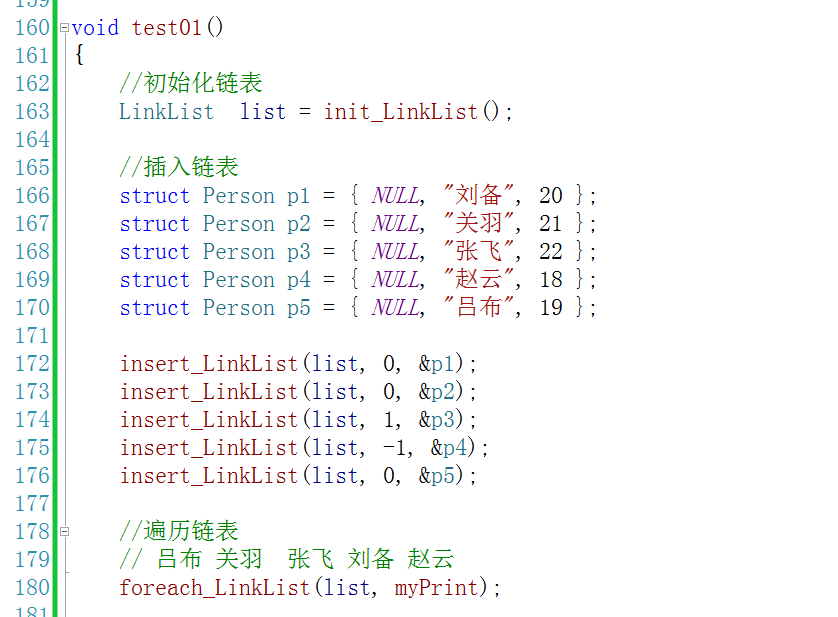
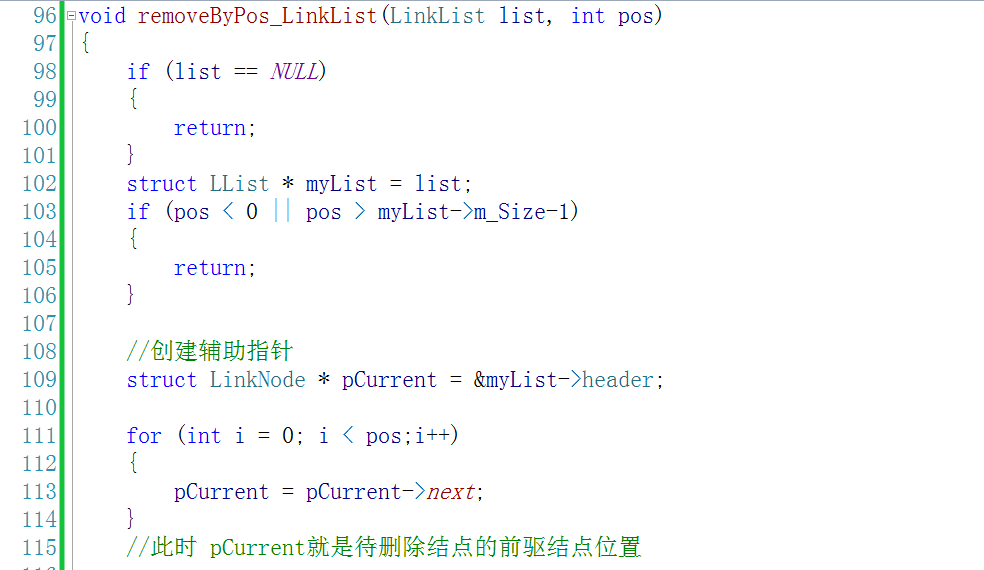
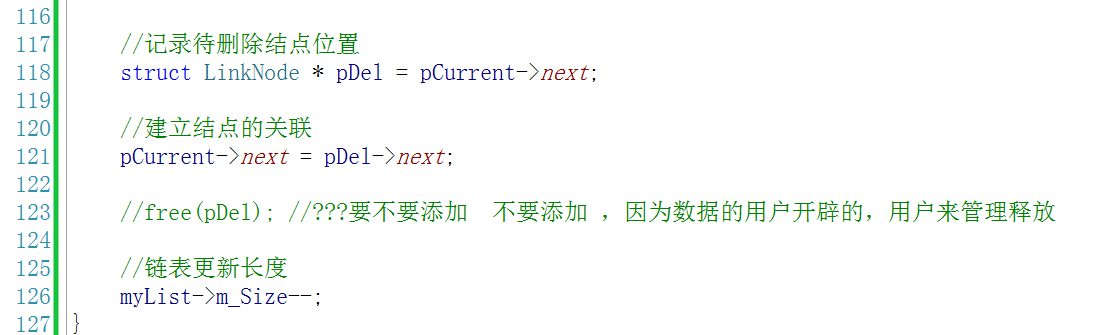
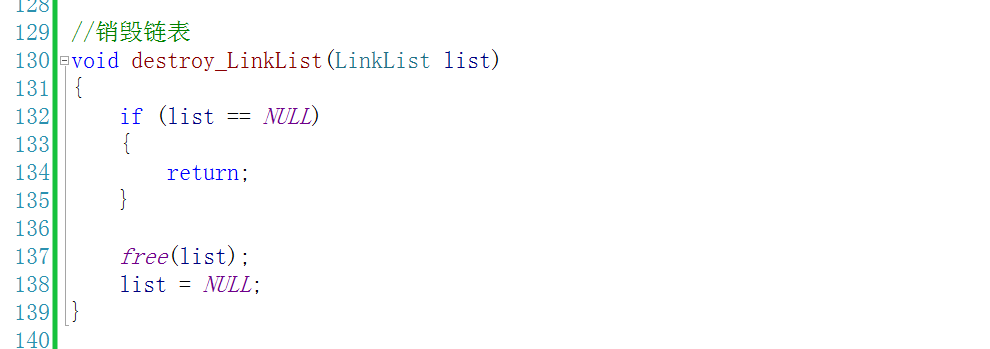
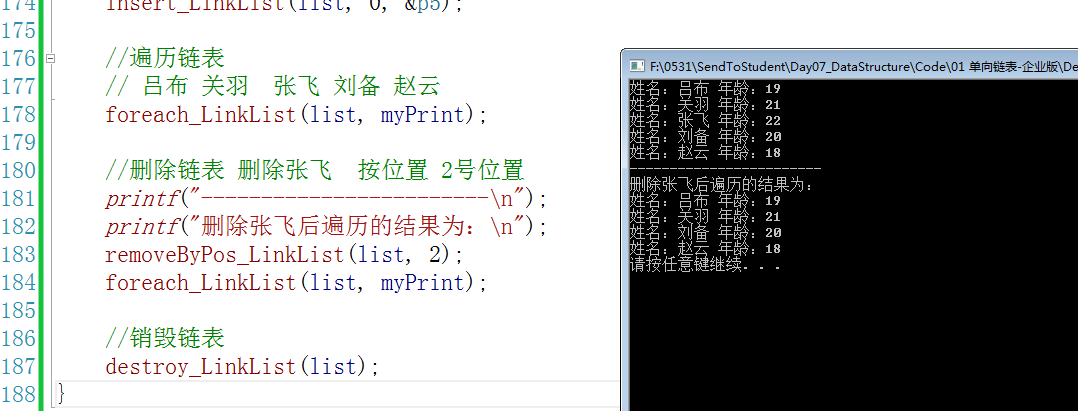
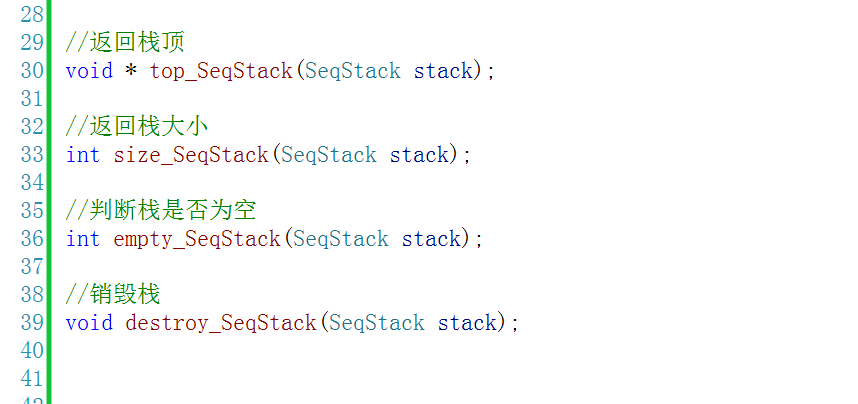
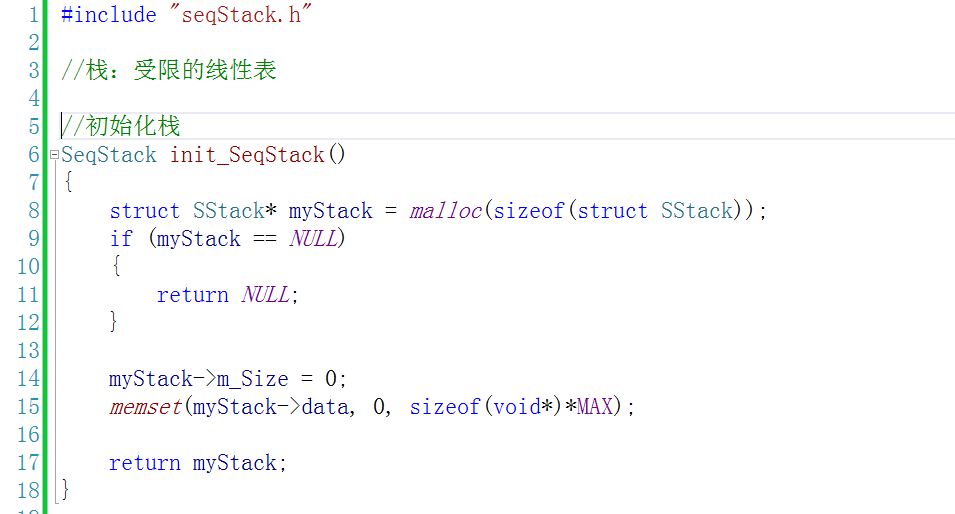
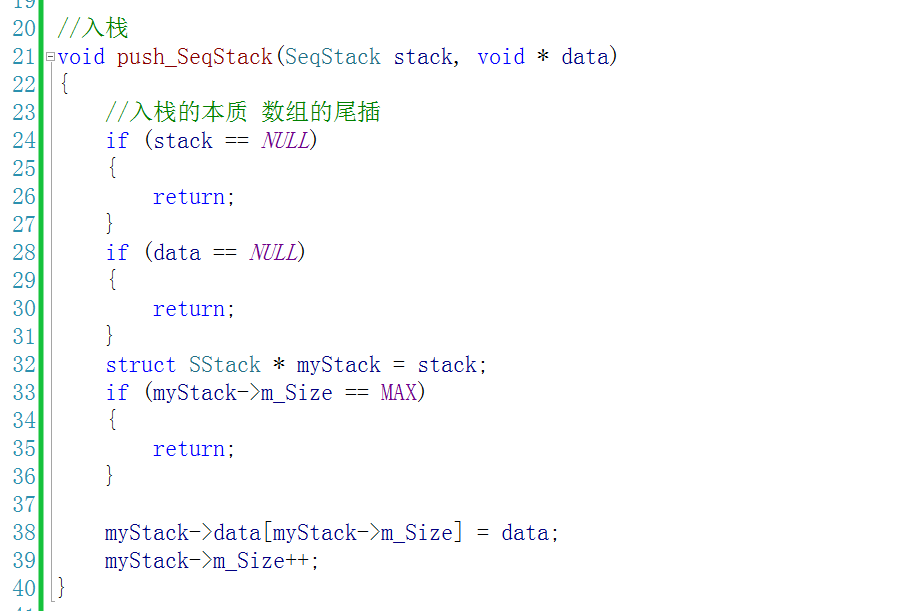
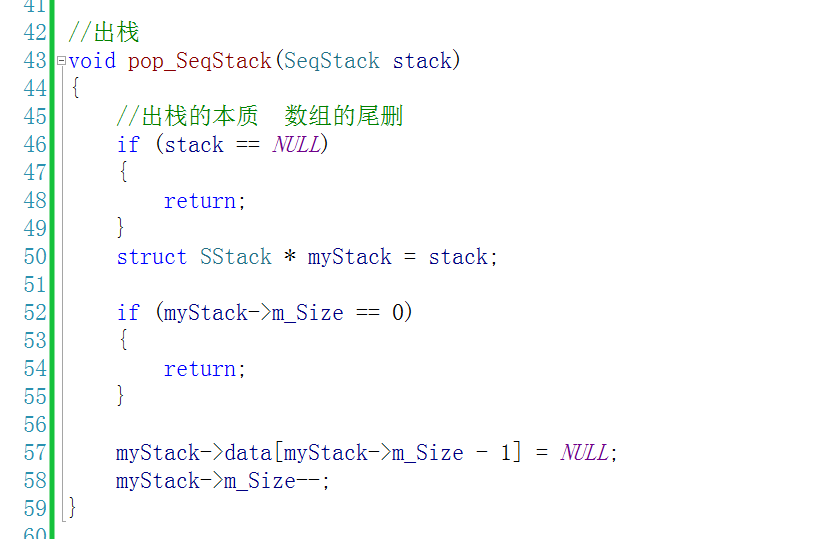
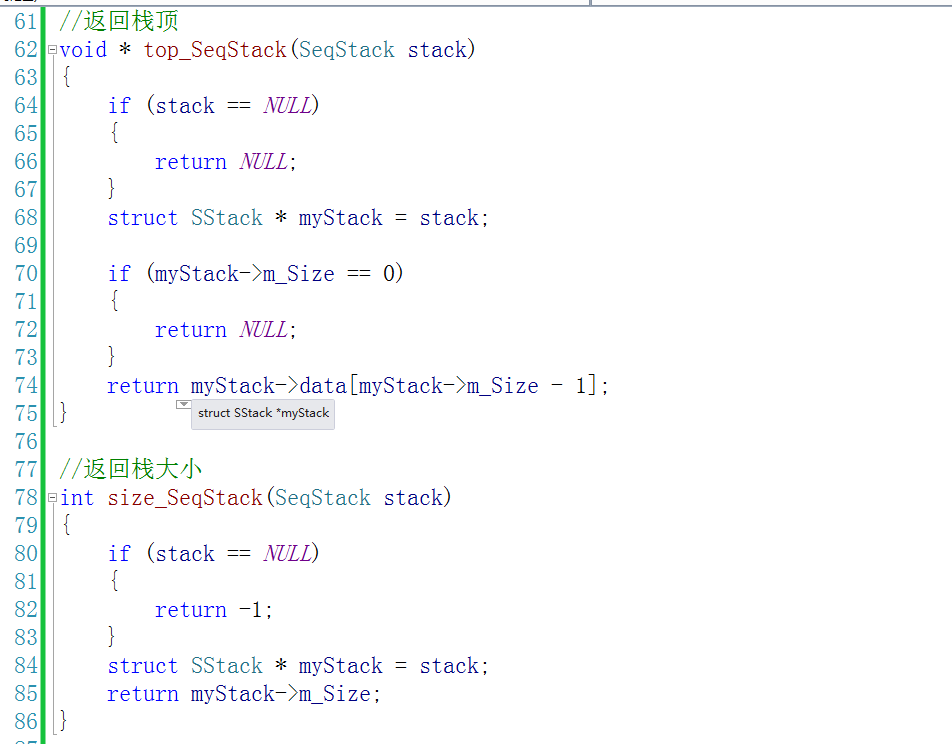
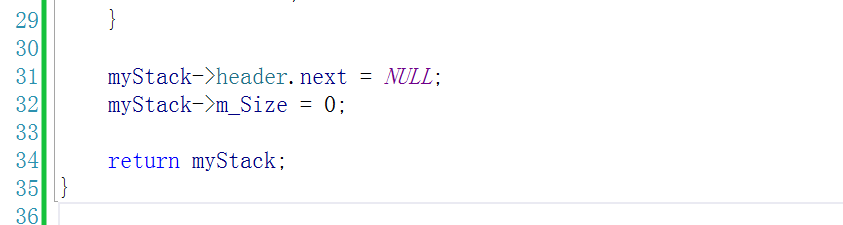
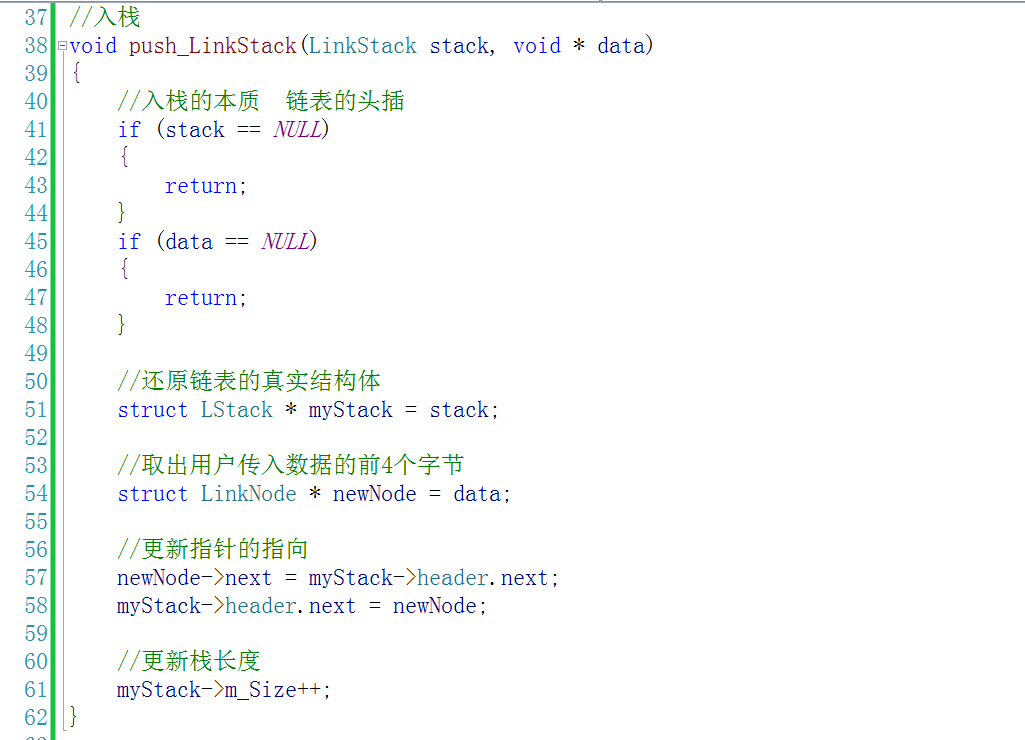
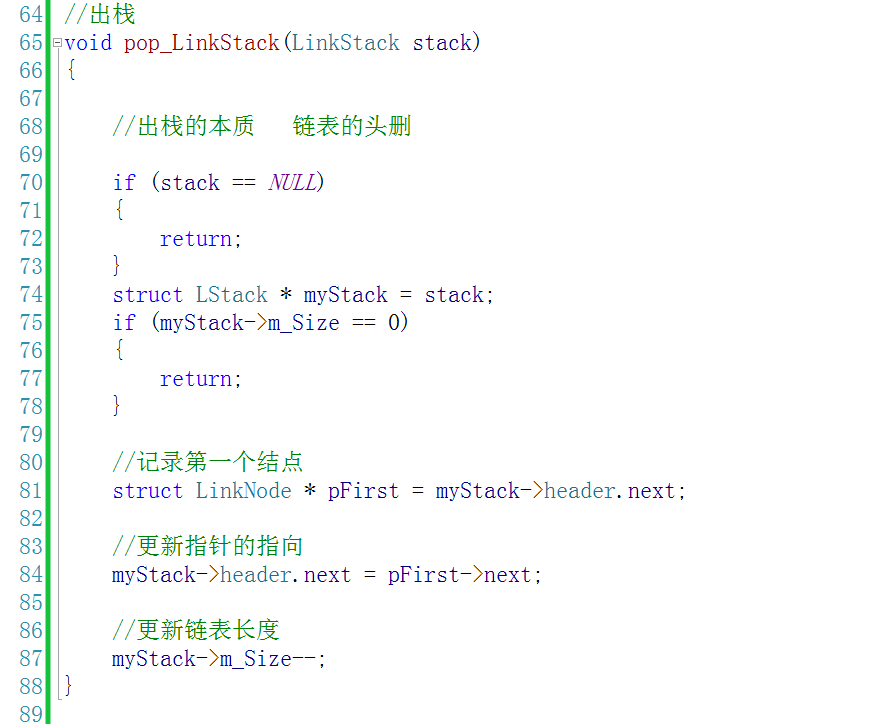
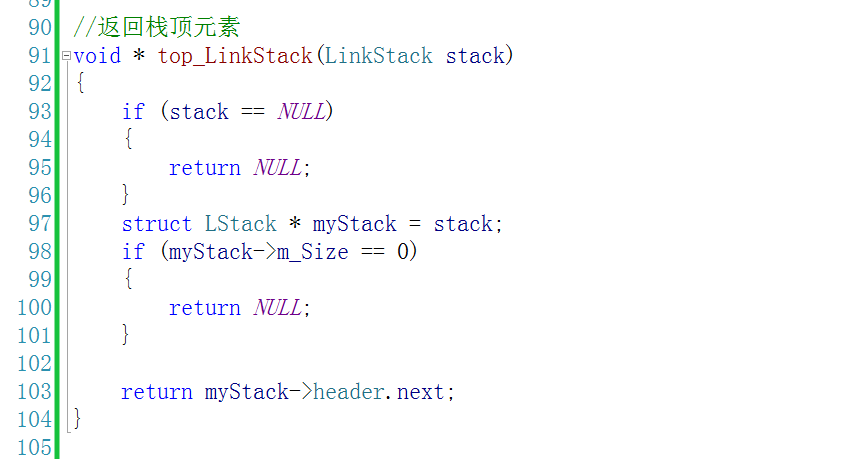
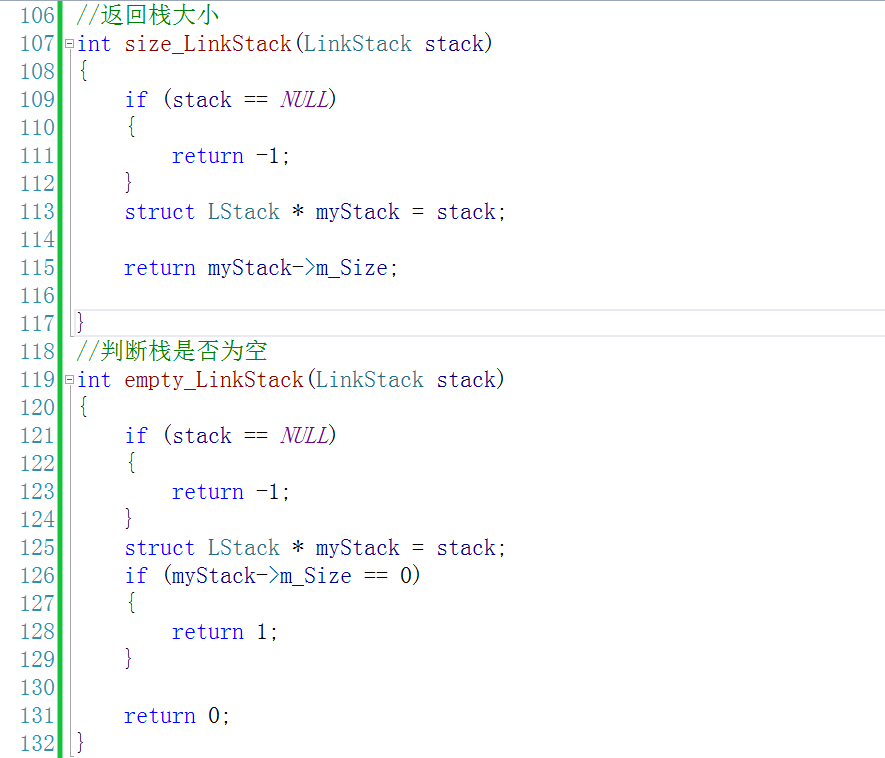
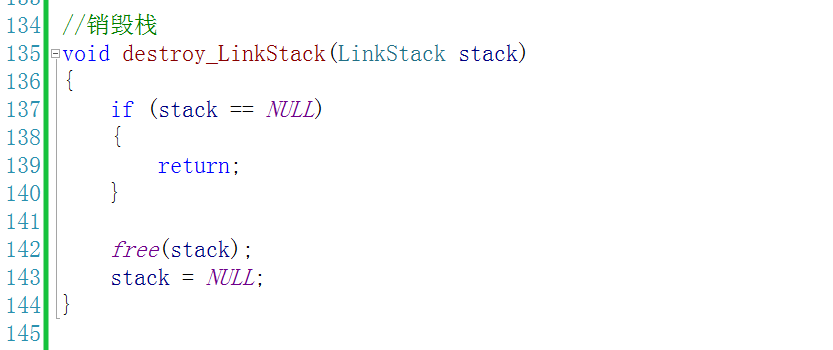
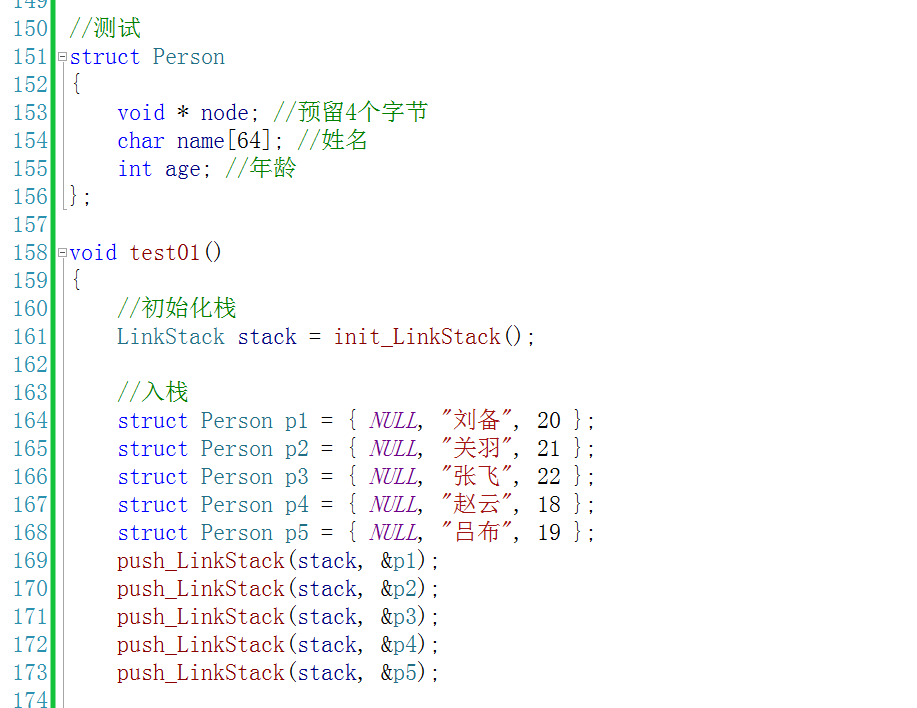
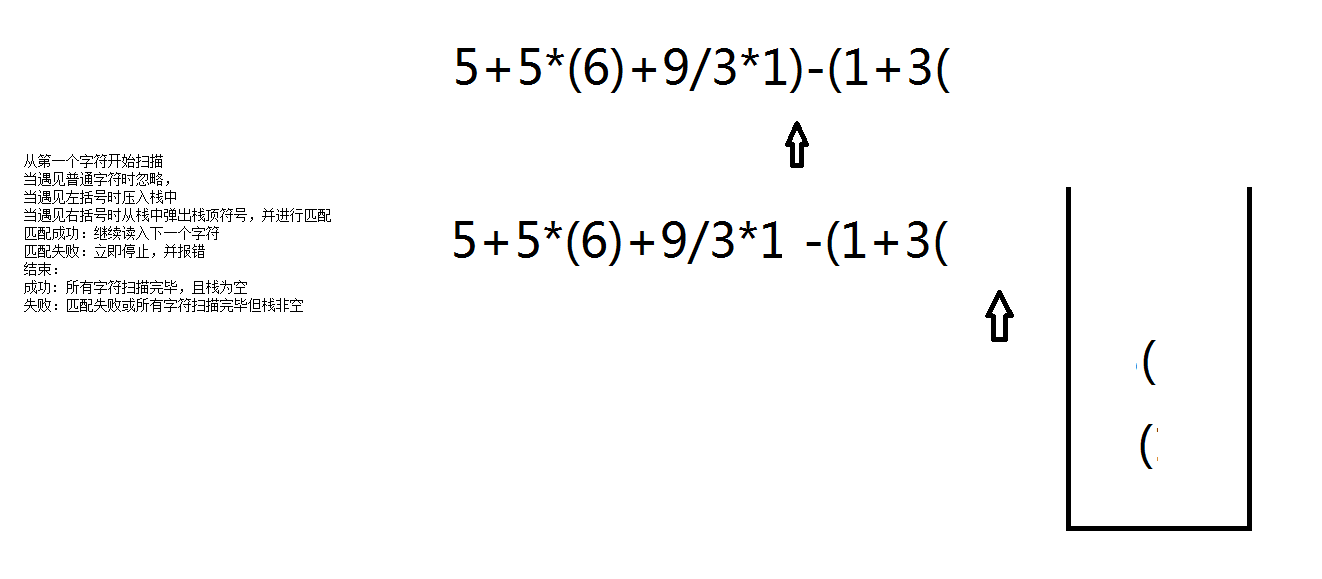
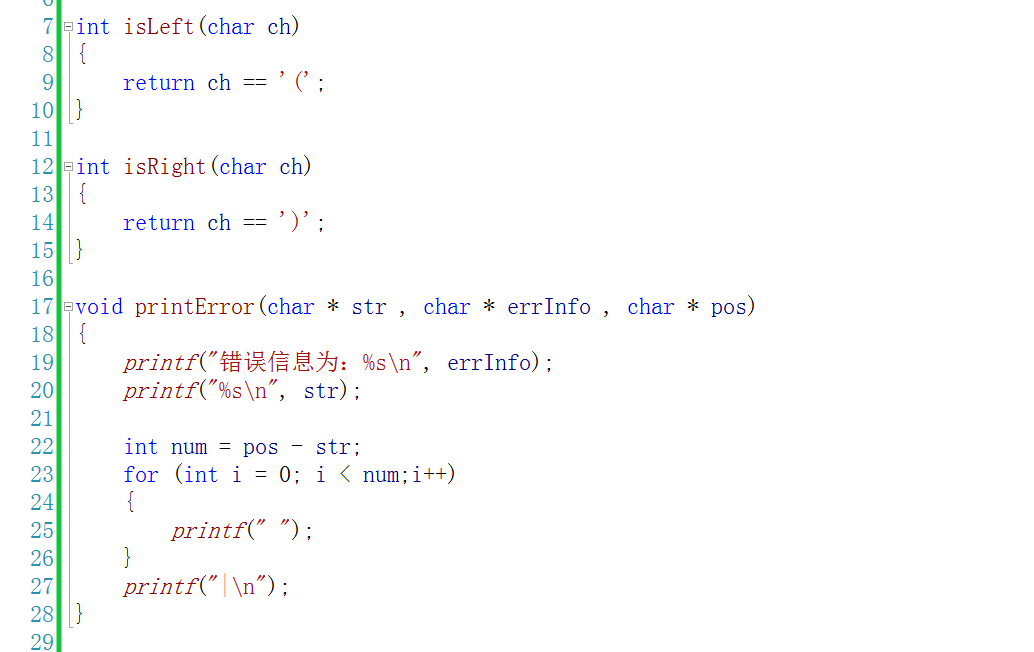
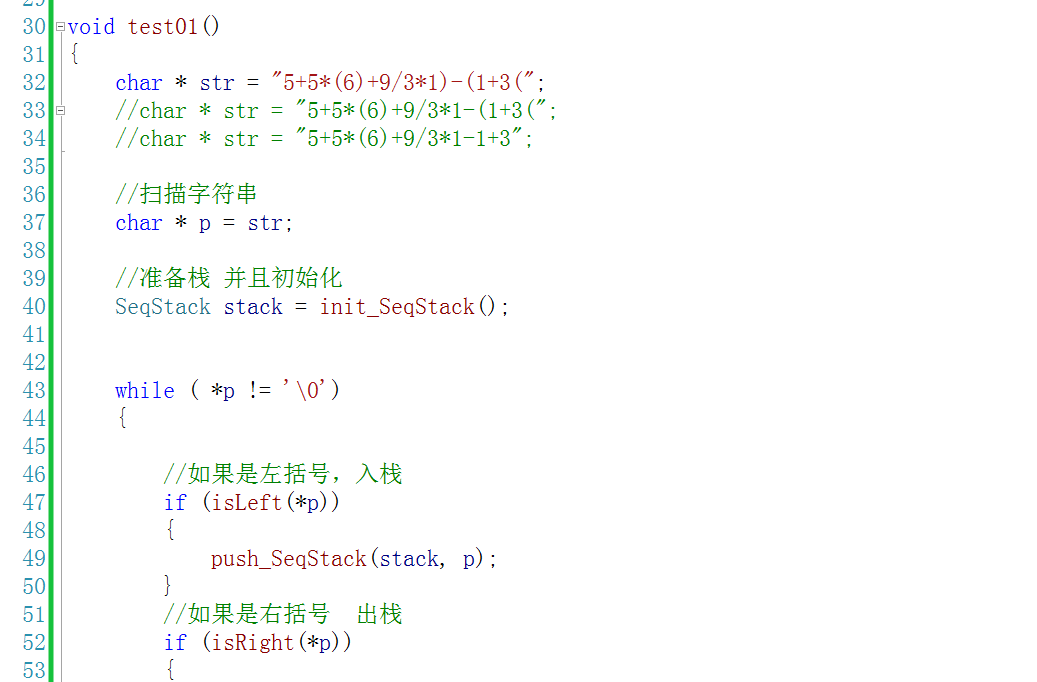
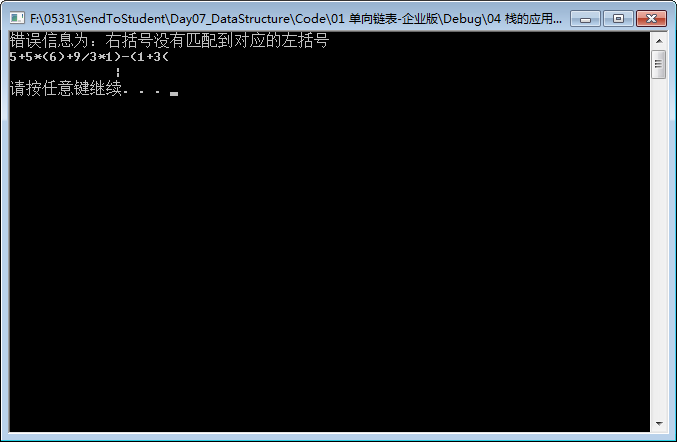
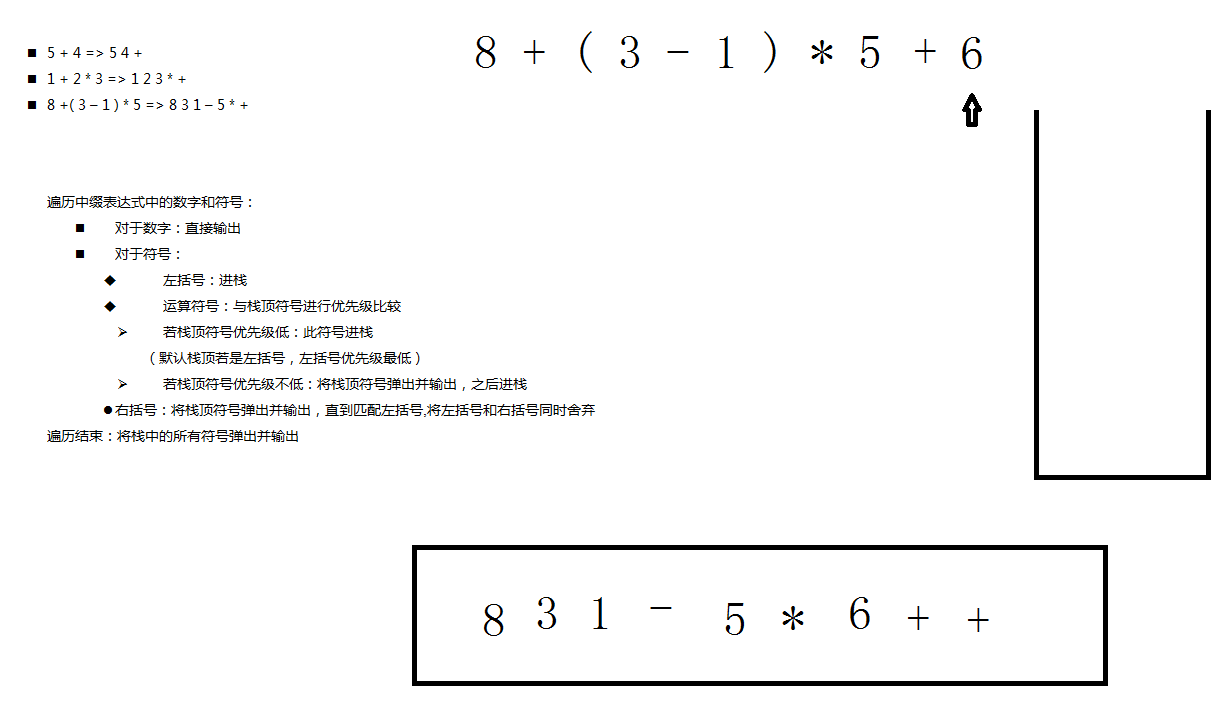
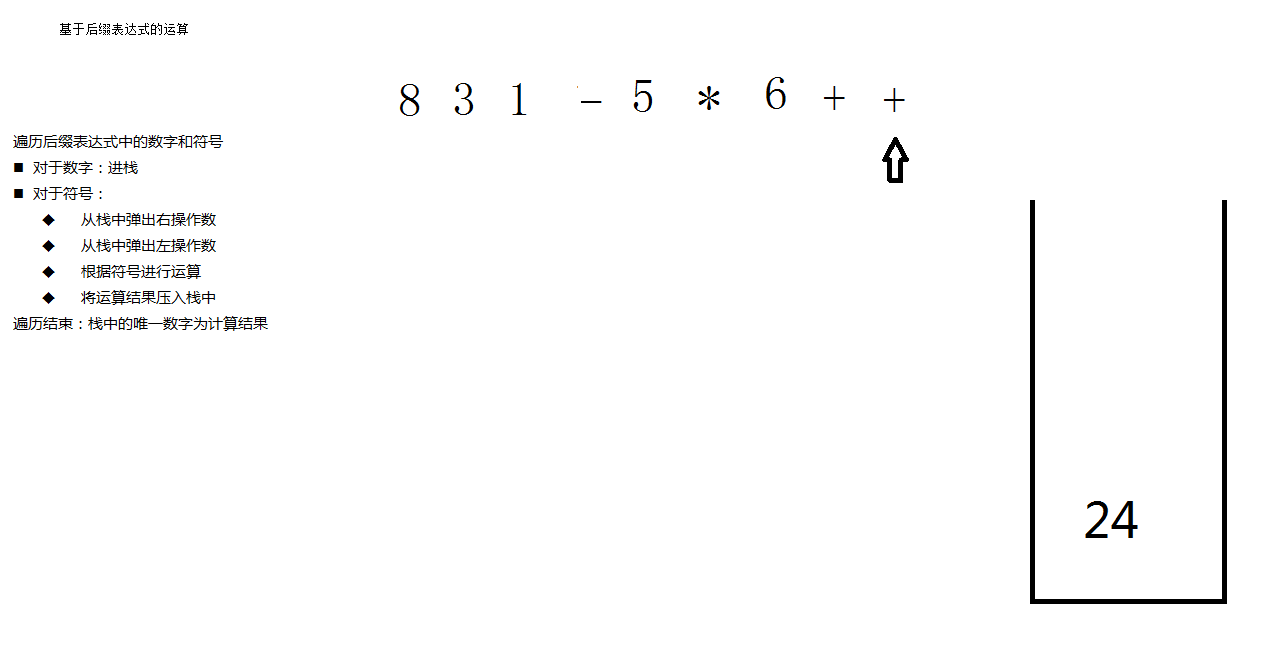
1. 单向链表-企业版
   1. 设计
      1. 结点结构体只维护指针域，不维护数据
      2. 
   2. 对外功能
      1. 初始化
         1. 
      2. 插入
         1. 
         2. 
      3. 遍历
         1. 
      4. 测试
         1. 
         2. 
      5. 删除链表
         1. 
         2. 
      6. 销毁链表
         1. 
      7. 测试
         1. 
2. 栈的顺序存储
   1. 符合先进后出的数据结构
   2. 利用数组模拟栈的结构
      1. 
      2. 
      3. 
      4. 
      5. 
      6. 
      7. 
      8. 测试
      9. 
      10. 
3. 栈的链式存储
   1. 利用链表模拟栈的结构
   2. 
   3. 
   4. 
   5. 
   6. 
   7. 
   8. 
   9. 测试
   10. 
   11. 
4. 栈的应用案例-就近匹配
   1. 
   2. 算法思路
      1. 从第一个字符开始扫描
      2. 当遇见普通字符时忽略，
      3. 当遇见左括号时压入栈中
      4. 当遇见右括号时从栈中弹出栈顶符号，并进行匹配
      5. 匹配成功：继续读入下一个字符
      6. 匹配失败：立即停止，并报错
      7. 结束：
      8. 成功: 所有字符扫描完毕，且栈为空
      9. 失败：匹配失败或所有字符扫描完毕但栈非空
   3. 实现
      1. 
      2. 
      3. 
      4. 
5. 中缀表达式和后缀表达式
   1. 中缀转后缀
      1. 
   2. 基于后缀进行运算
      1. 
6. 队列
   1. 符合 先进先出的数据结构
   2. 