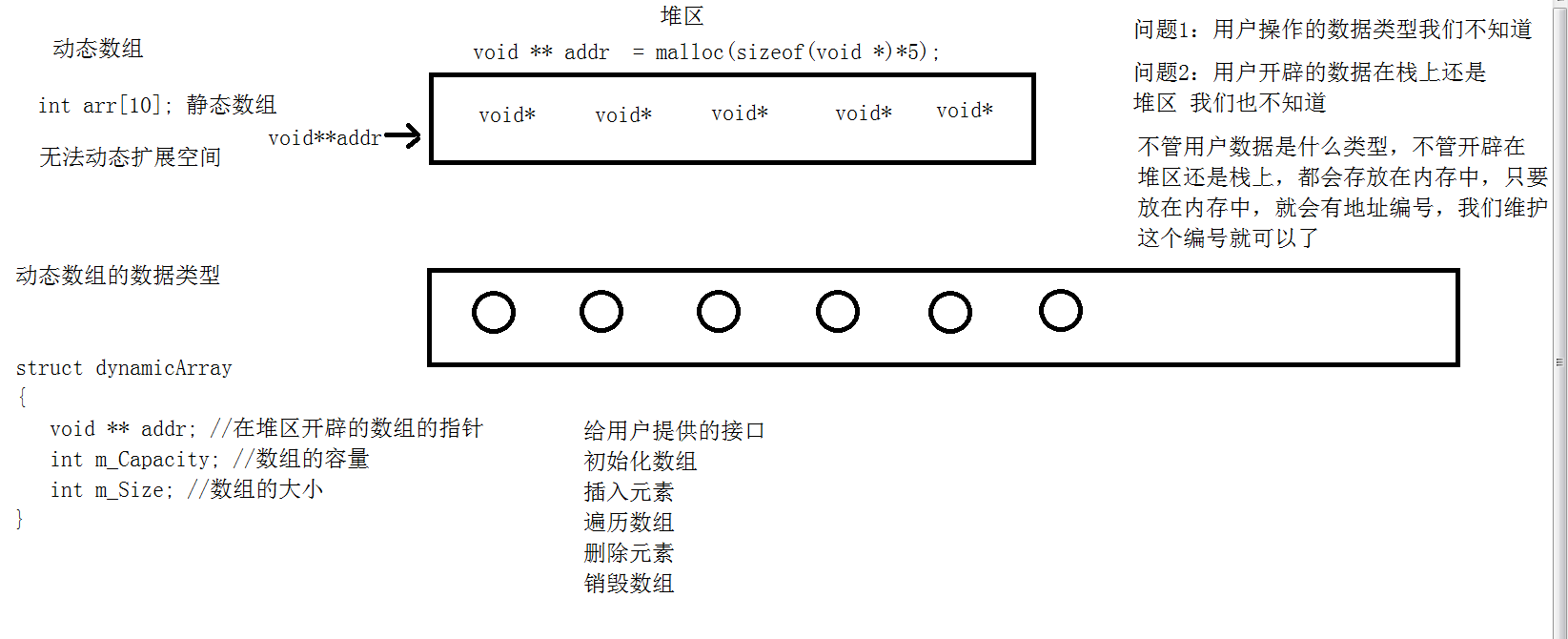
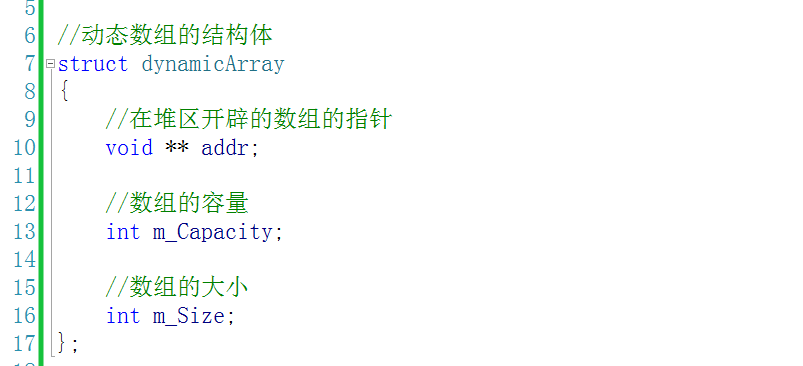
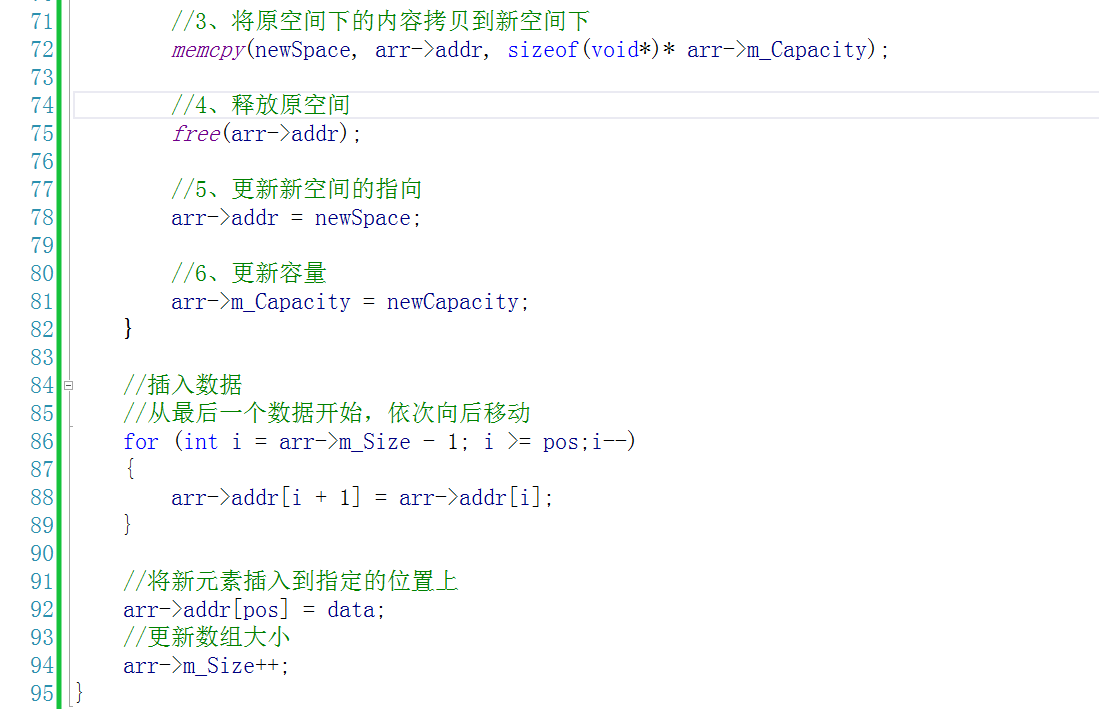
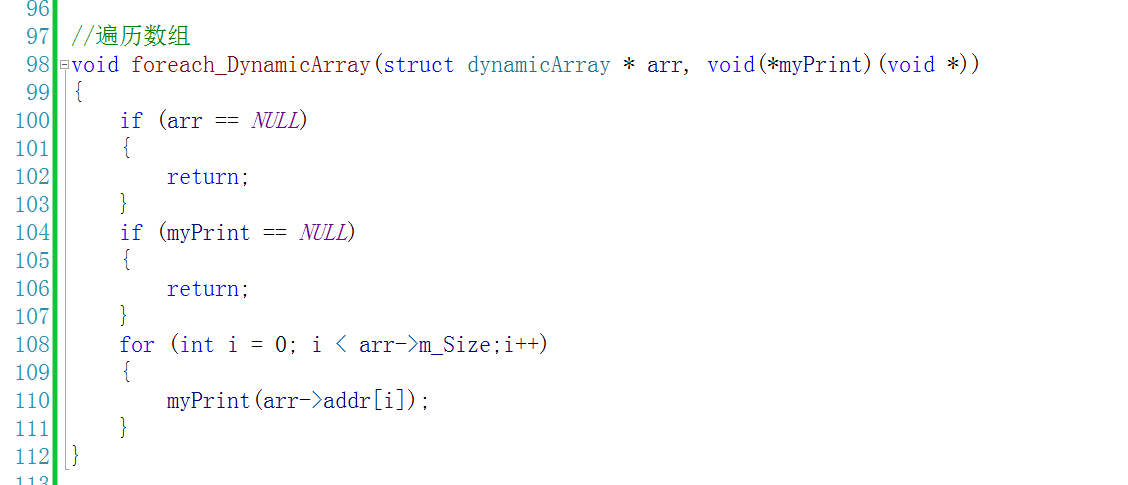
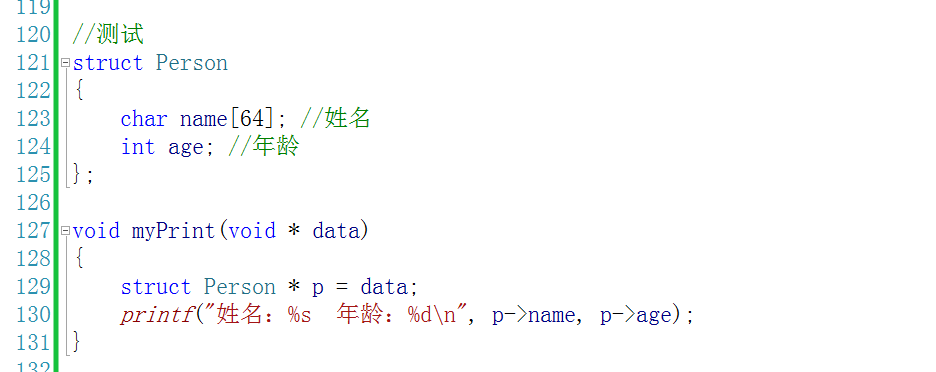
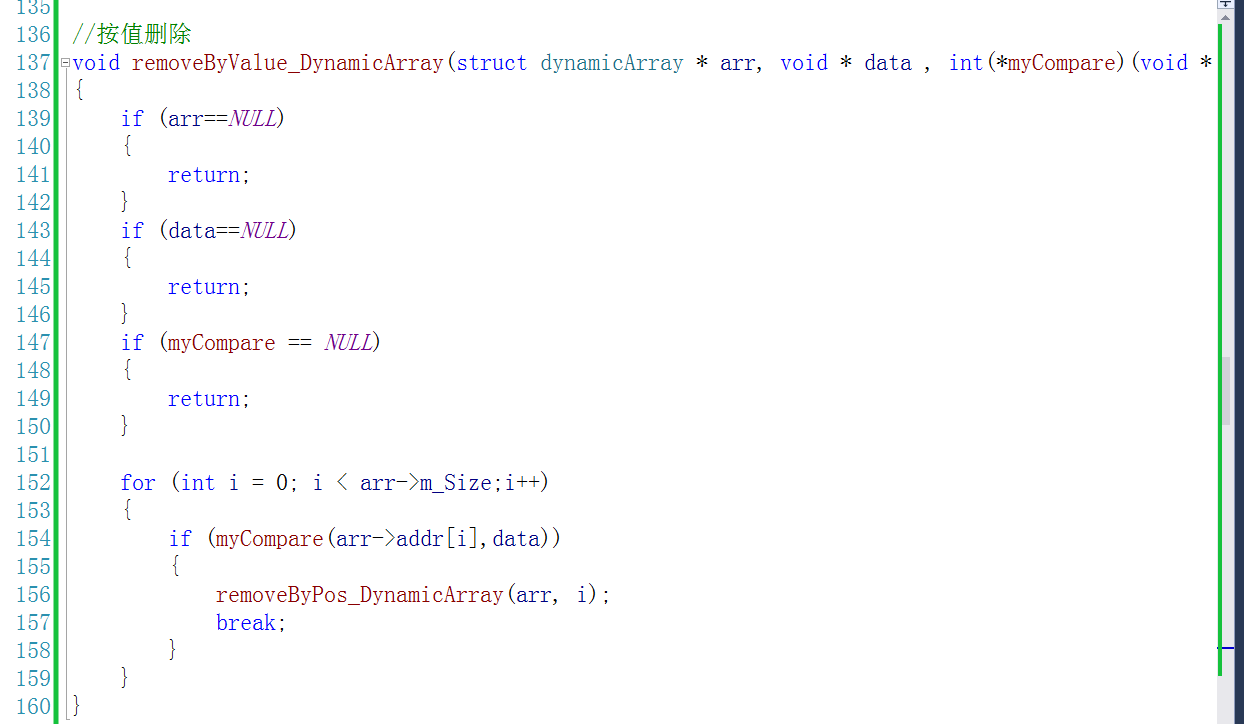
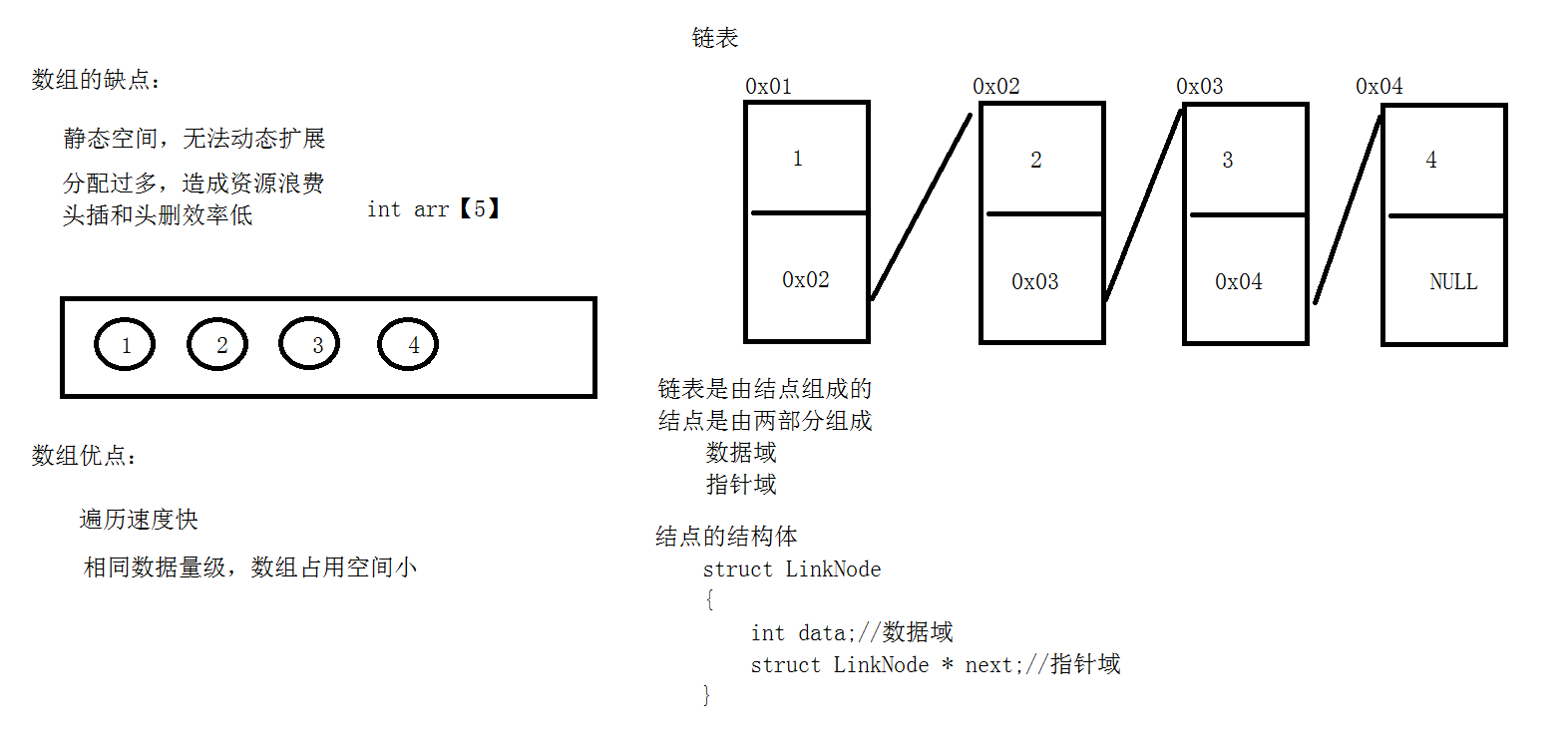
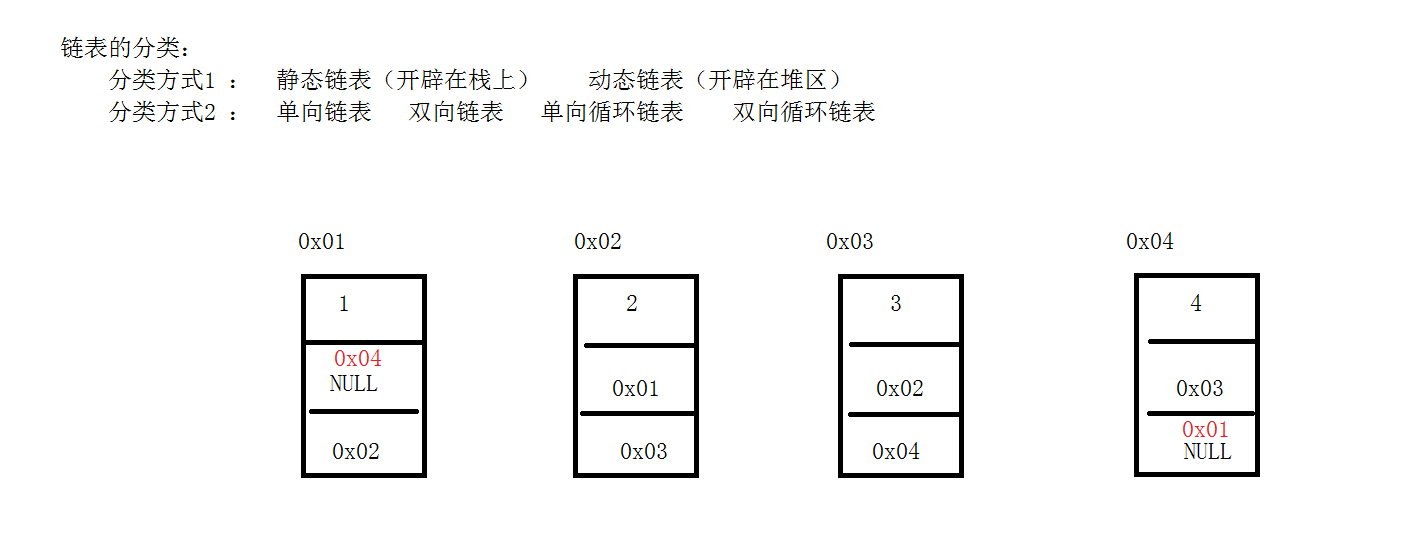
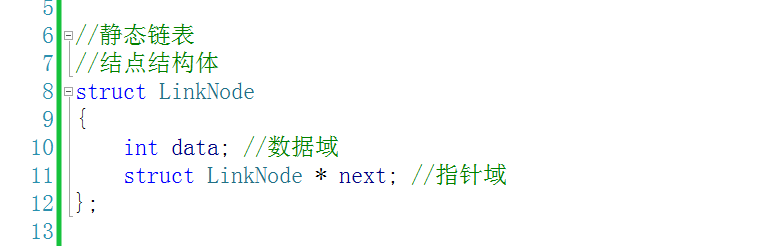
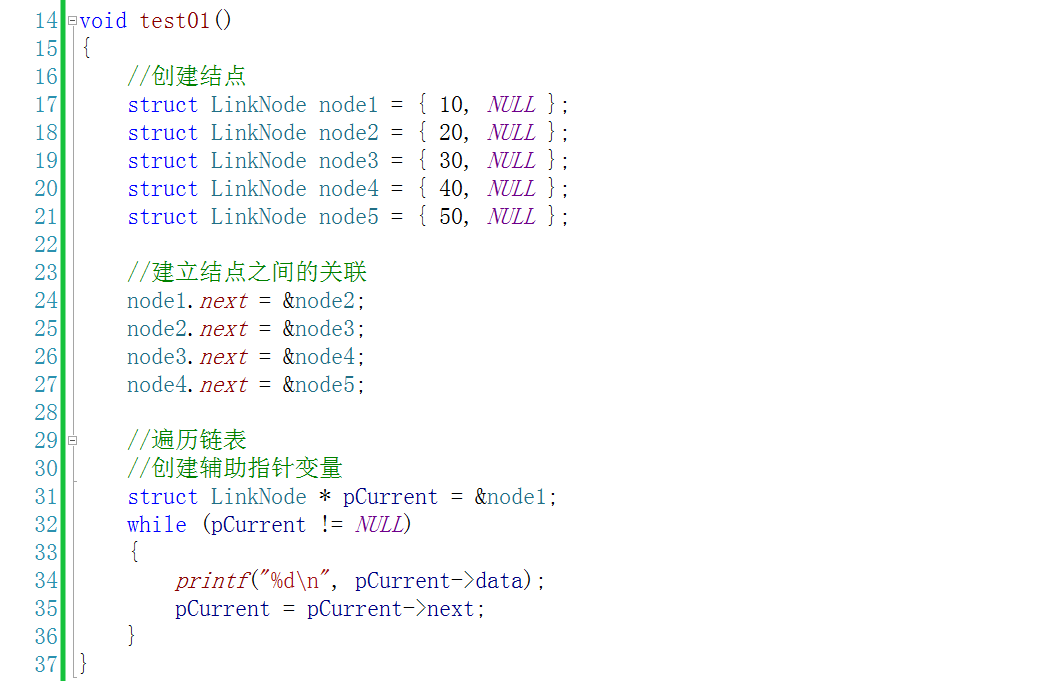
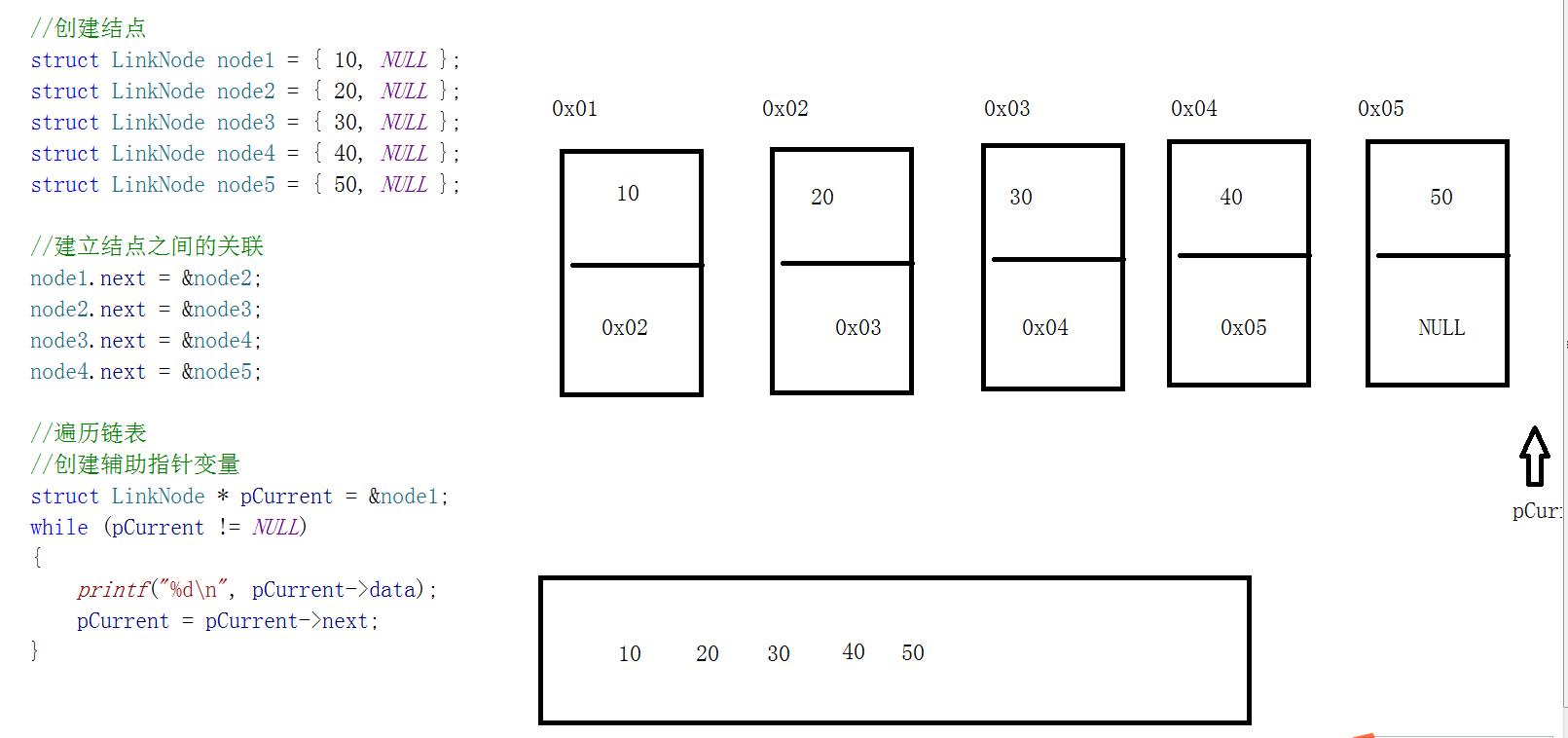
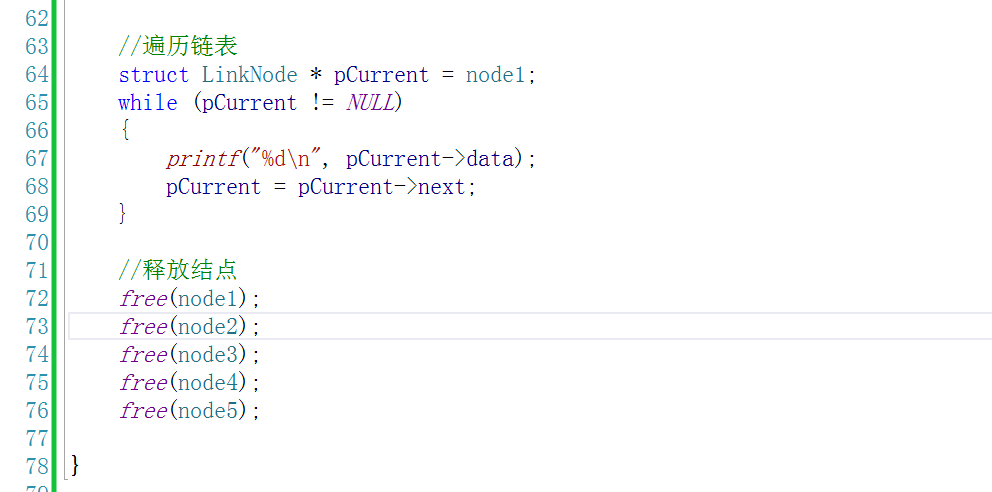
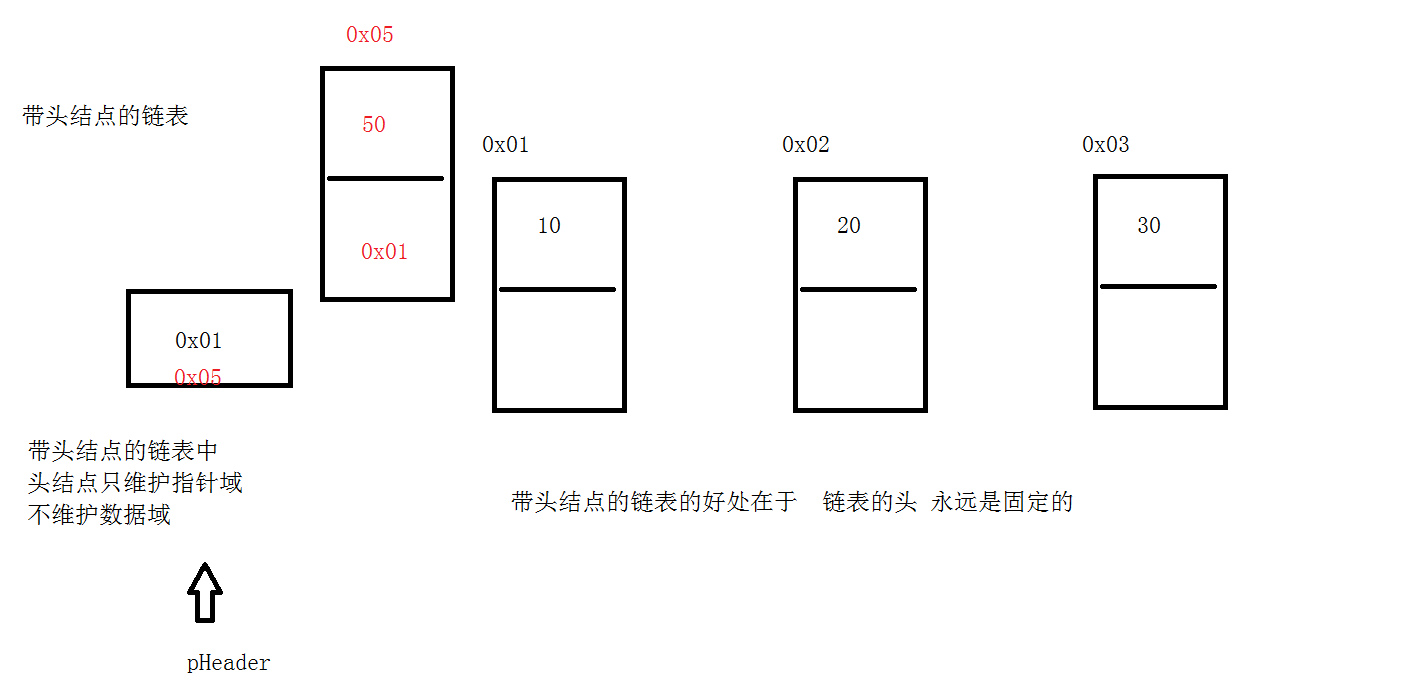
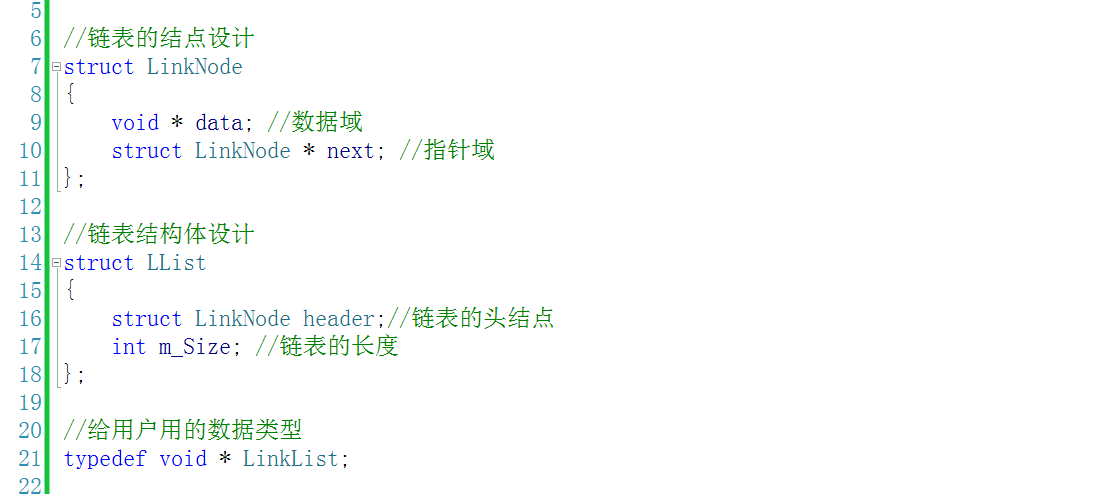
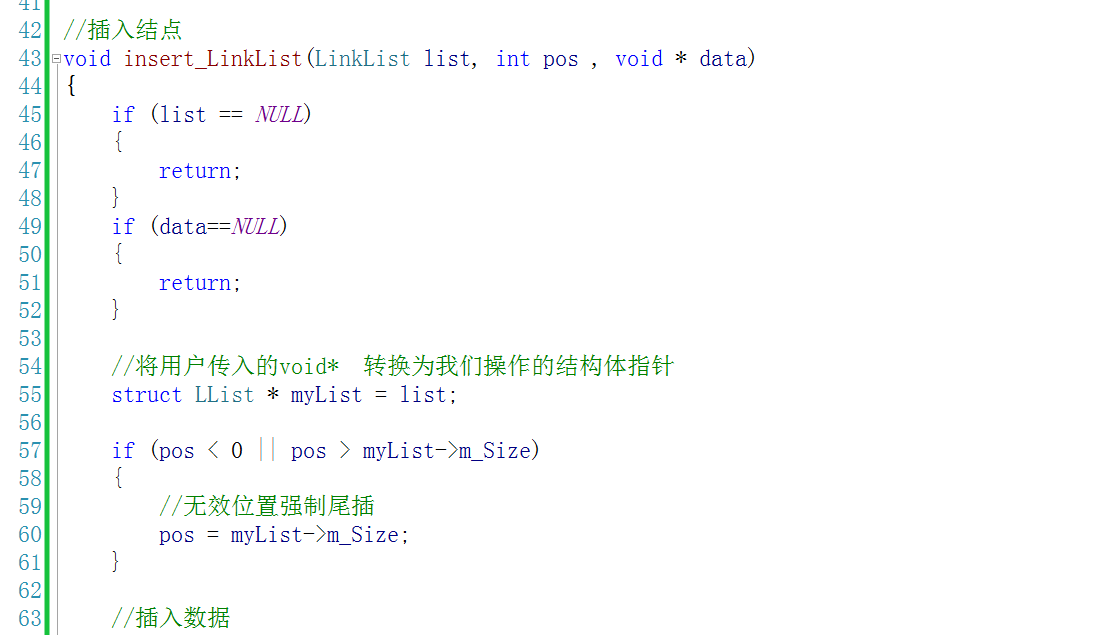
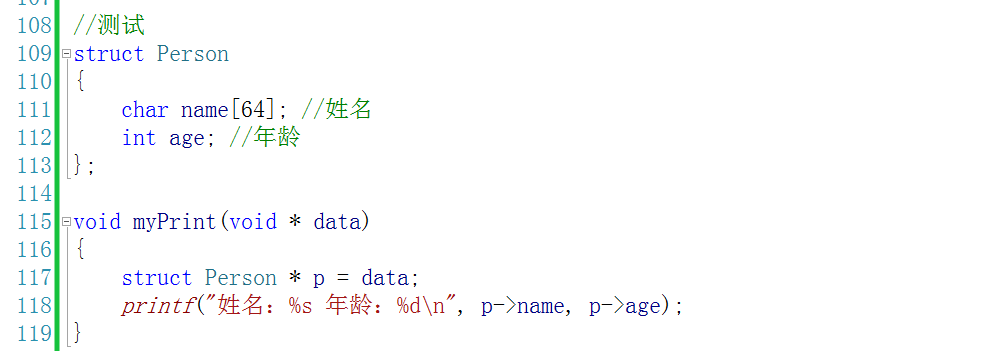
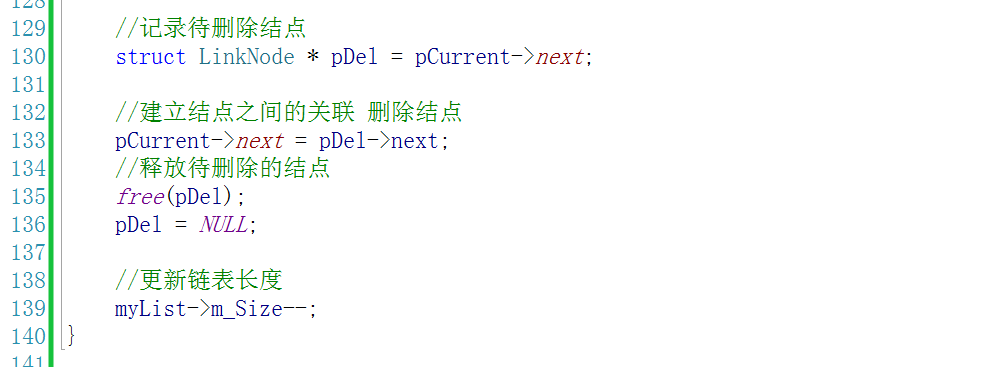
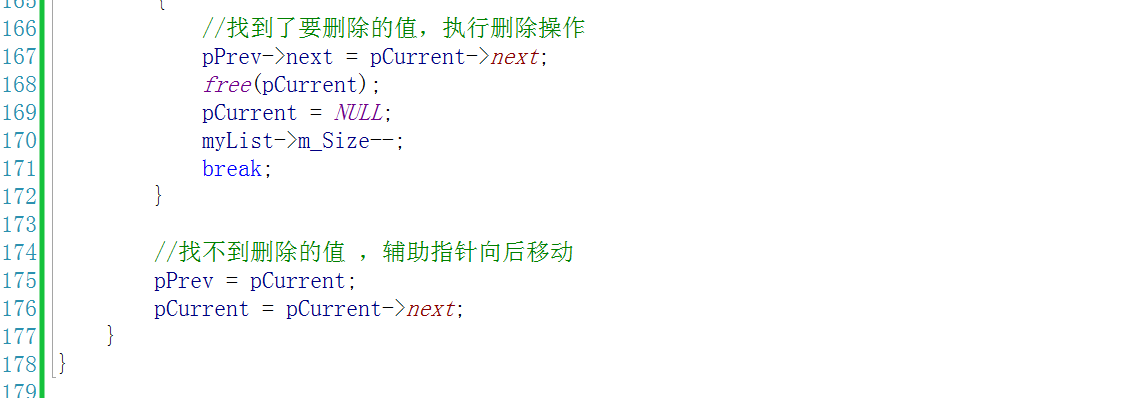
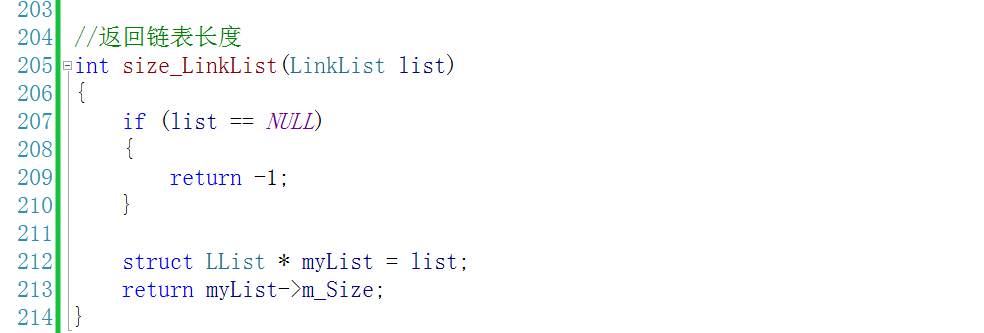
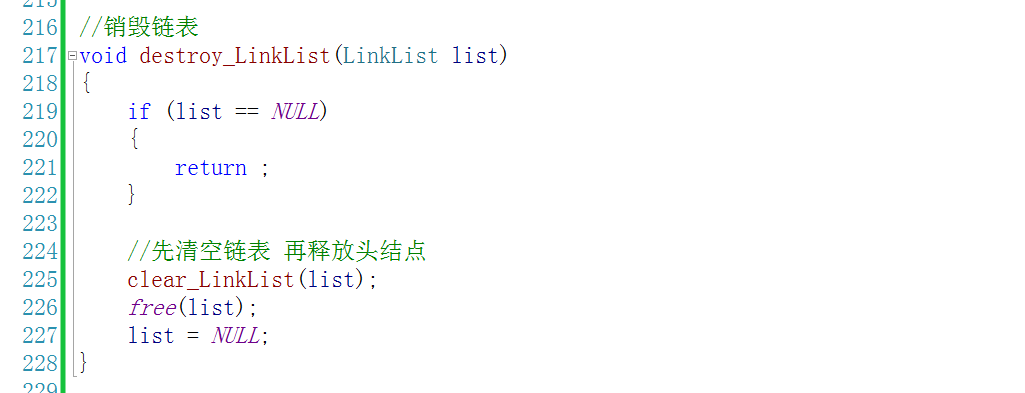
1. 数据结构概述
   1. 算法的五个特性：
      1. 输入 输出 有穷 确定 可行
   2. 数据结构分类
      1. 逻辑结构
         1. 集合结构
         2. 线性结构 1:1
         3. 树形结构 1:n
         4. 图形结构 n:n
      2. 物理结构
         1. 顺序存储
         2. 链式存储
2. 动态数组
   1. 设计
      1. 
   2. 结构体设计
      1. 
   3. 初始化功能
      1. 
   4. 插入数组
      1. 
      2. 
   5. 遍历数组
      1. 
   6. 测试
      1. 
      2. 
   7. 删除元素
      1. 按位置删除
         1. 
      2. 按值删除
         1. 
      3. 销毁数组
         1. 
   8. 测试
      1. 
   9. 实现分文件编写 方便复用
      1. 创建dynamicArray.h和 dynamicArray.c文件
      2. 头文件中留下 结构体声明 和 函数声明
      3. 源文件中留下 函数实现
      4. 在main文件中，包含dynamicArray.h头文件 就可以使用动态数组了
3. 链表的基本概念
   1. 链表是由 结点组成的
   2. 结点是由 两部分组成
      1. 数据域
      2. 指针域
   3. 设计结点结构体
      1. struct LinkNode
      2. {
         1. int data; 数据域
         2. struct LinkNode\* next;指针域
      3. }
   4. 
4. 链表的分类
   1. 静态链表（栈） 动态链表（堆）
   2. 单向链表 、双向链表、单向循环链表、双向循环链表
      1. 
5. 静态链表和动态链表
   1. 
   2. 
   3. 
   4. 
   5. 
6. 带头结点的链表
   1. 好处：永远固定了链表的头
   2. 头结点 只维护指针域 不维护数据域
   3. 
7. 单向链表实现
   1. 
   2. 
   3. 
   4. 
   5. 遍历链表
   6. 
   7. 测试
      1. 
      2. 
   8. 删除链表
      1. 按位置删除
         1. 
         2. 
      2. 按值删除
         1. 
         2. 
         3. 
   9. 清空链表
      1. 
   10. 返回链表长度
       1. 
   11. 销毁链表
       1. 
   12. 测试
       1. 