1. 数据结构基本理论
   1. 算法五个特性：
      1. 输入、输出、有穷、确定、可行
   2. 数据结构分类
      1. 逻辑结构：集合、线性、树形、图形
      2. 物理结构：顺序存储、链式存储
2. 动态数组实现
   1. 设计
      1. struct dynamicArray
      2. 属性：
      3. void \*\* pAddr 维护真实在堆区创建的数组的指针
      4. int m\_capacity; 数组容量
      5. int m\_size; 数组大小
   2. 动态数组初始化
      1. struct dynamicArray \* init\_DynamicArray(int capacity)
   3. 插入数组
      1. void insert\_DynamicArray(struct dynamicArray \* array , int pos , void \* data)
   4. 遍历数组
      1. void foreach\_DynamicArray(struct dynamicArray \* array, void(\*myPrint)(void\*))
   5. 删除数组
      1. 按照位置删除
      2. void removeByPos\_DynamicArray(struct dynamicArray \* array , int pos)
      3. 按照值删除
      4. void removeByValue\_DynamicArray(struct dynamicArray \* array , void \* data , int (\* myCompare)(void \* ,void \*))
   6. 销毁数组
      1. void destroy\_DynamicArray(struct dynamicArray\* array)
   7. 实现分文件编写
3. 单向链表
   1. 设计
      1. struct LinkNode 节点结构体
      2. struct LList 链表结构体
      3. typedef void \* LinkList 给用户使用链表指针
   2. 初始化链表
      1. LinkList init\_LinkList()
   3. 插入链表
      1. void insert\_LinkList(LinkList list, int pos, void \* data)
   4. 遍历链表
      1. void foreach\_LinkList(LinkList list, void(\*myForeach)(void \*))
   5. 删除链表
      1. 按照位置void removeByPos\_LinkList(LinkList list, int pos)
      2. 按照值void removeByValue\_LinkList(LinkList list , void \* data , int(\*myCompare)(void \* ,void \*) )
   6. 清空
      1. void clear\_LinkList(LinkList list)
   7. 返回链表长度
      1. int size\_LinkList(LinkList list)
   8. 销毁
      1. void destroy\_Linklist(LinkList list)