**数据结构与算法 第一次实验**

学号：22920212204392 姓名：黄勖

1. **实验目的**
2. 了解线性表的基础实现方法与原理，理解线性表的基本操作的代码编写方式
3. 学会灵活按照线性表的存储内容自由编写线性表的存储结构
4. 在线性表的基础上进而理解链表的基础实现方法，理解链表的基本操作的代码编写方式
5. 通过实验探索线性表与链表的相似点与区别，发现二者在操作实现上的不同
6. **实验内容**

1-1 设链表的存储结构如下：

typedef struct Node

{

Type data; //数据域；Type: 用户定义数据类型

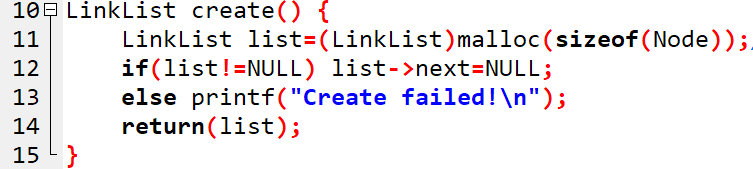
struct Node \*next; //指针域

} Node, \*LinkList;

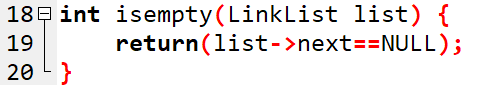
实现链表的基本操作。

在本次实验中实现的内容：

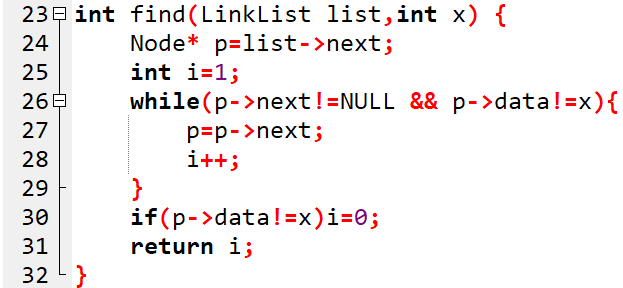
* 创建空链表



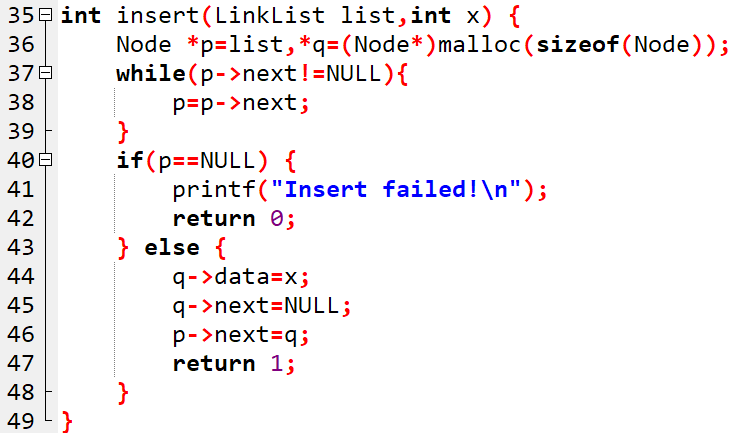
* 判断链表是否为空



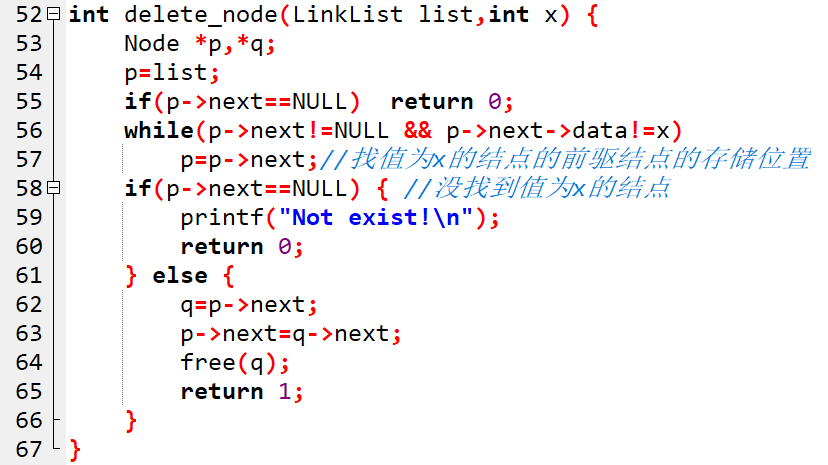
* 在链表中找某元素的存储位置



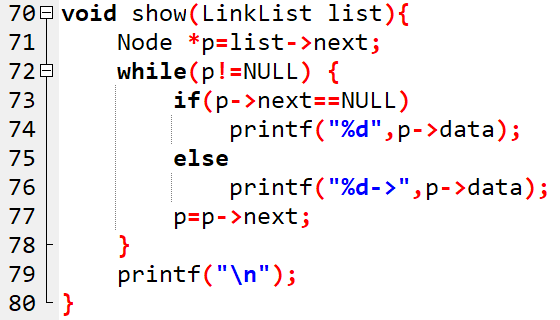
* 单链表的插入（链表末尾插入元素x）



* 单链表的删除（删除第一个值为x的结点）



* 打印链表

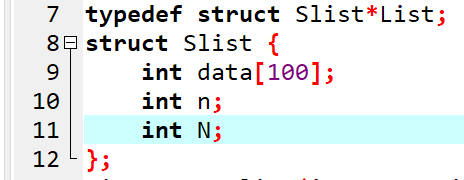


* Main函数中对编写操作的具体应用

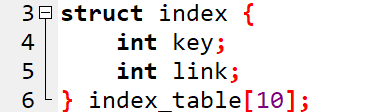
（编码具体内容见1-1.cpp）

1-2 实现顺序表的分级查找算法。基本要求包括：

1. **设计顺序表和索引表的存储结构。**
   1. 对于顺序表，用常见的方法存储：

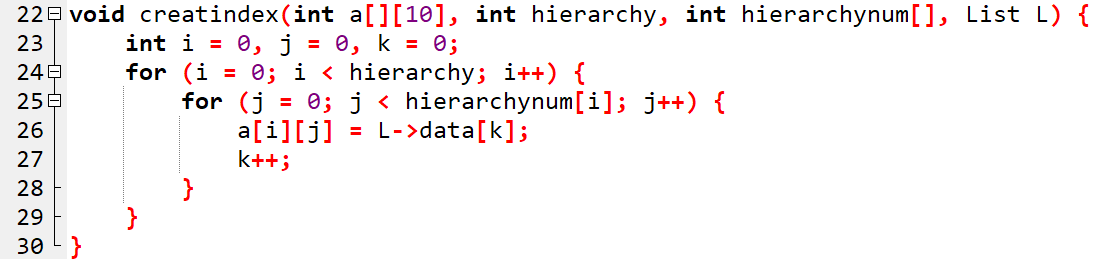


* 1. 对于索引表 创建key值与link值一一对应建立索引



1. **根据顺序表创建索引表。**

在实验过程中编写了如下函数创建索引表，其中List L为顺序表，通过循环将索引信息写入索引表a中，hierarchy与hierarchynum为分级的层次数量。



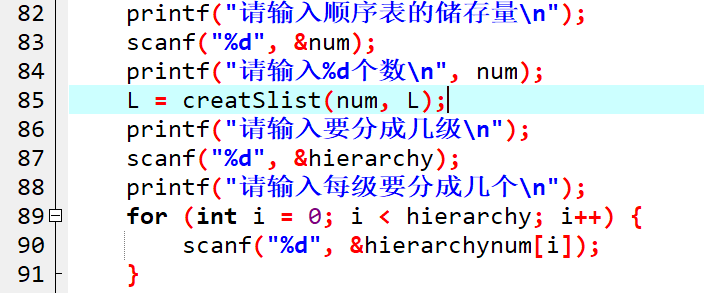
1. **设计分级查找算法。**

设计：要实现分级查找算法，首先要对存储的数据进行分级，并针对用户的分级建立对应的索引表，最后针对创建的索引表创建查找算法。

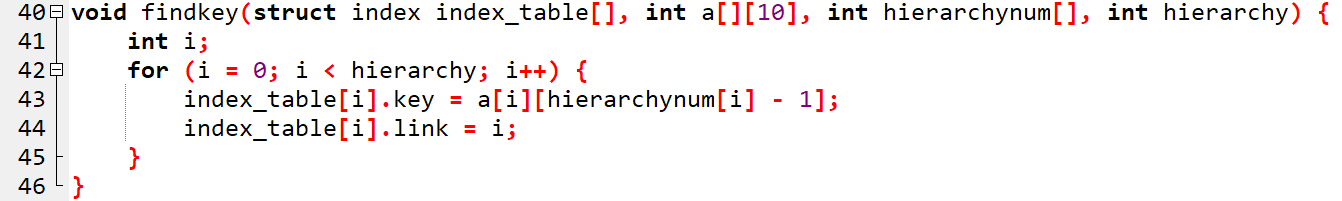
具体实现：

1. 根据级别（层次）数量创建索引表：其中hierarchy为级别总数，hierarchynum为每个级别中的元素个数

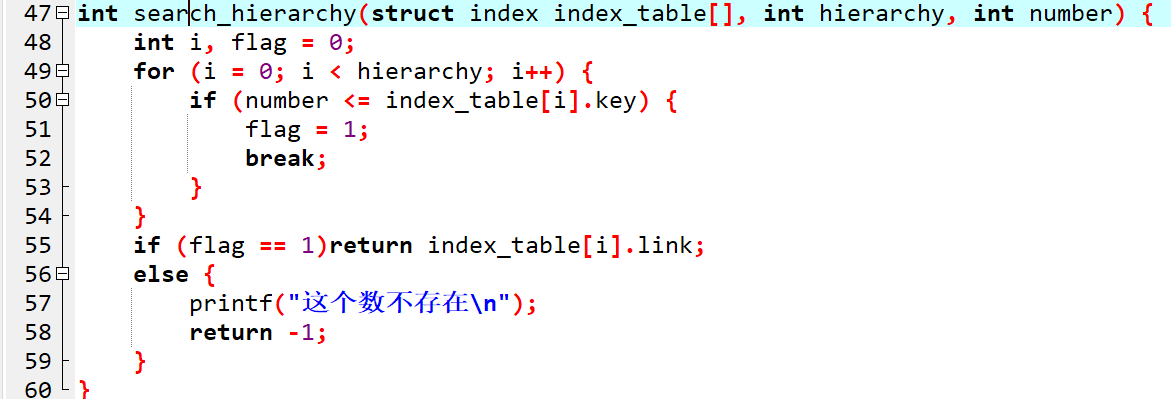
建立索引：



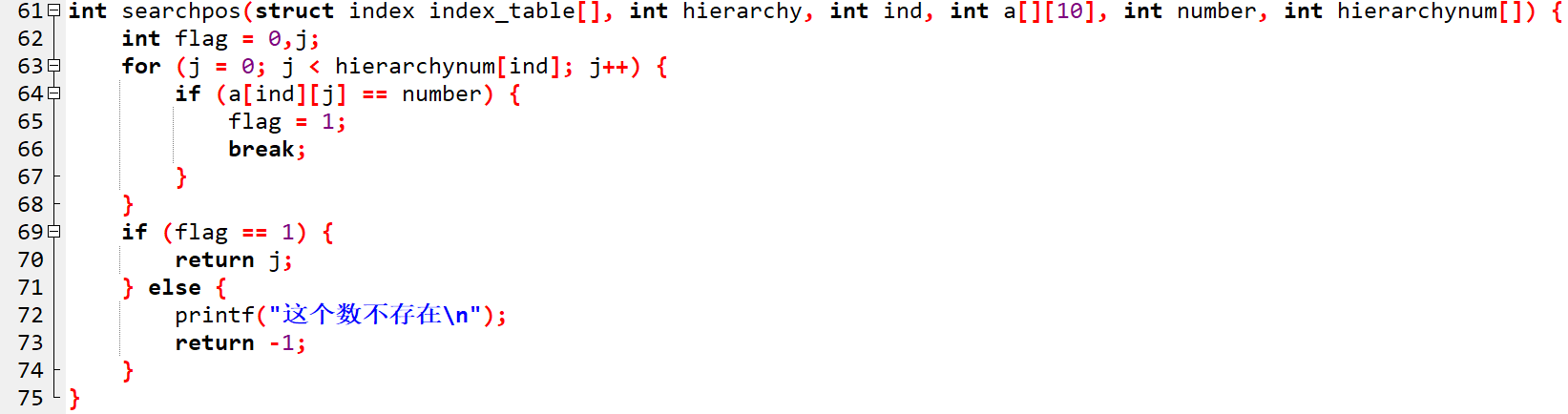
建立索引表：



1. 接下来需要编写的就是实现分级查找的功能
   1. 首先遍历查找级数



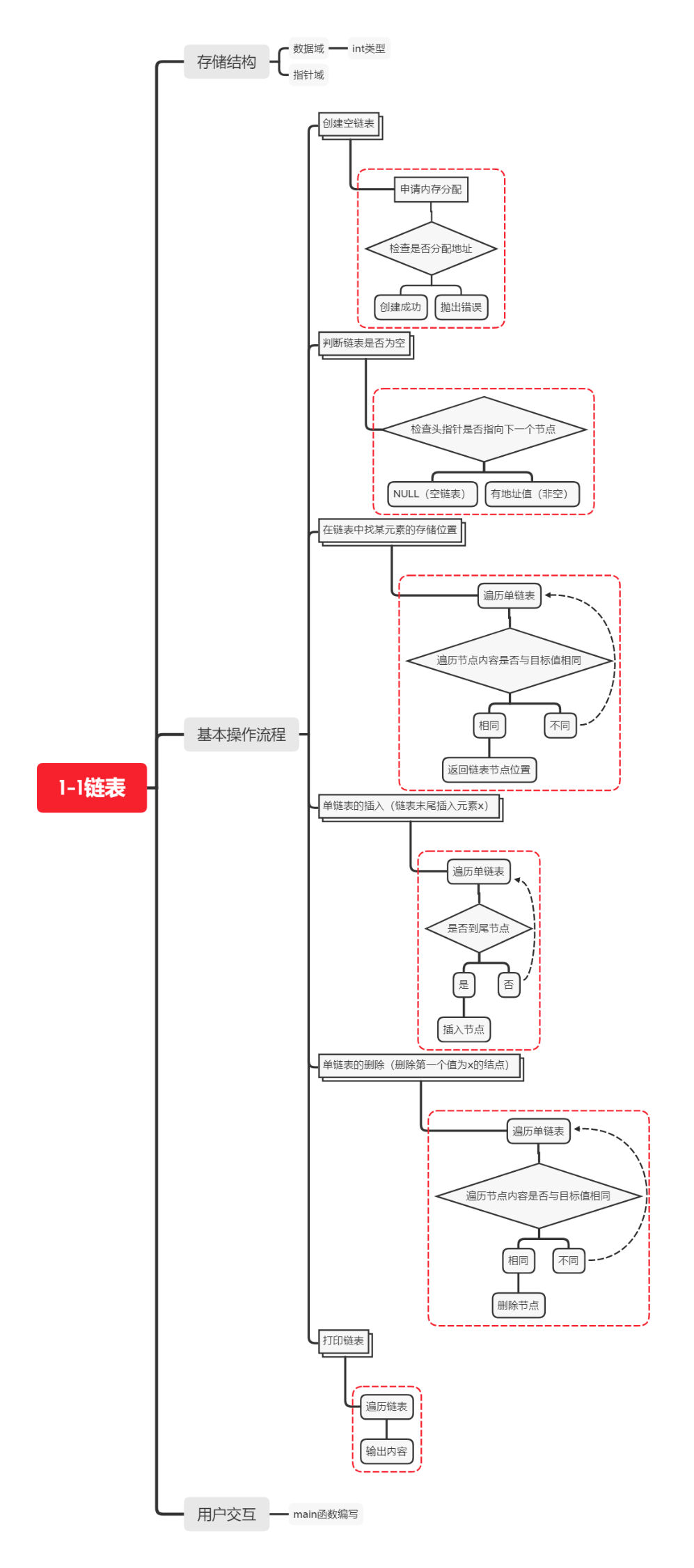
* 1. 最后再在这一级寻找需要的数据



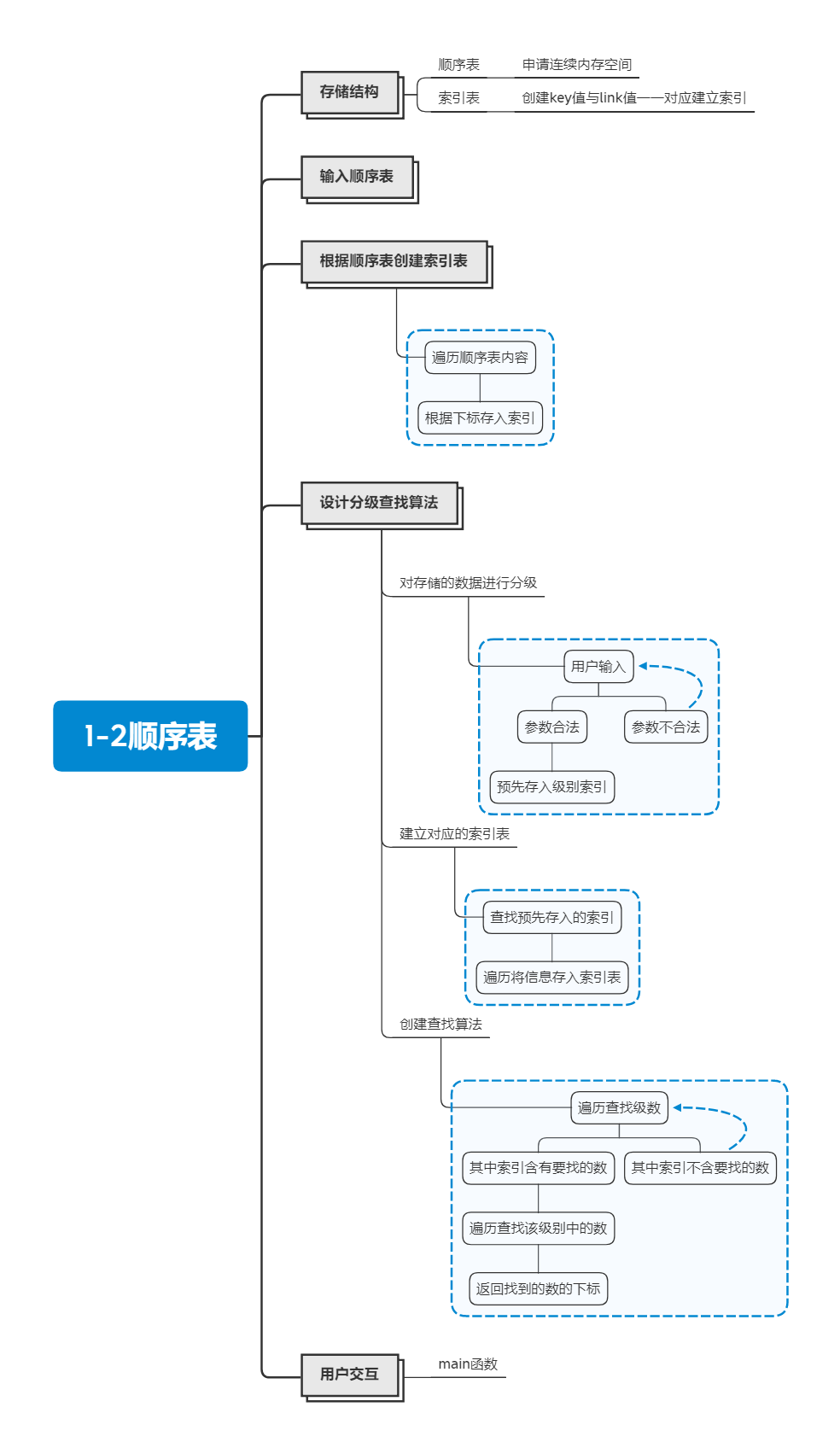
1. **最后根据编写的功能写出能与用户交互的main函数**
2. **主要算法流程图**

**1-1算法流程**

**（图片太小可见目录下的1-1链表.png）**

****

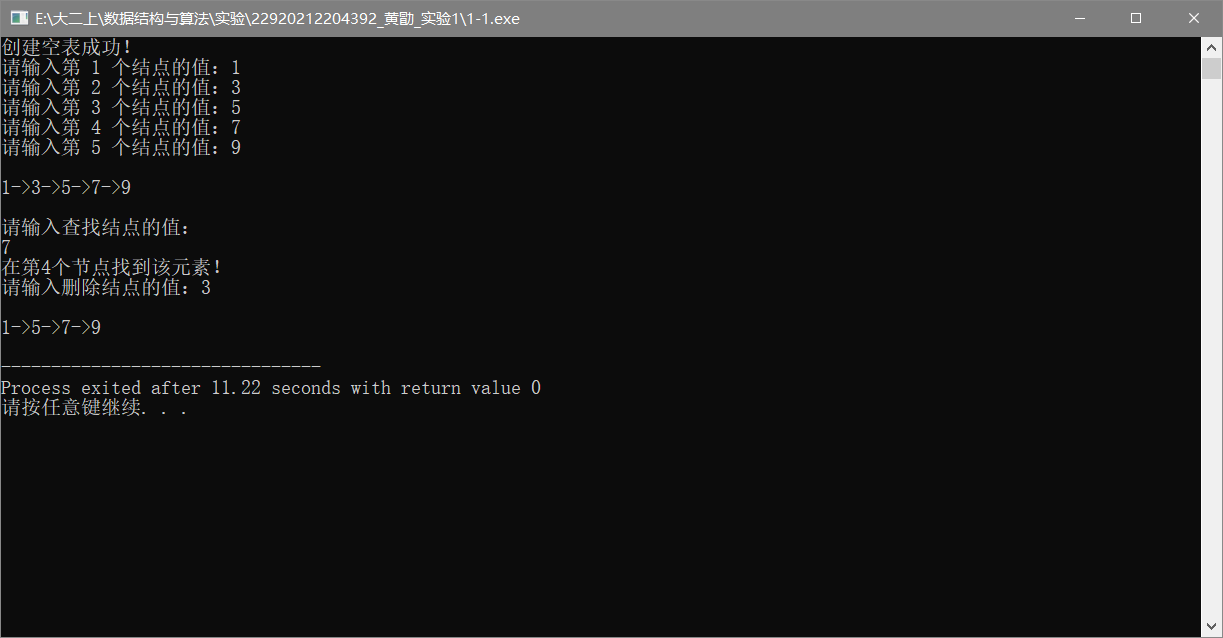
**1-2算法流程**

****

1. **实验结果：**

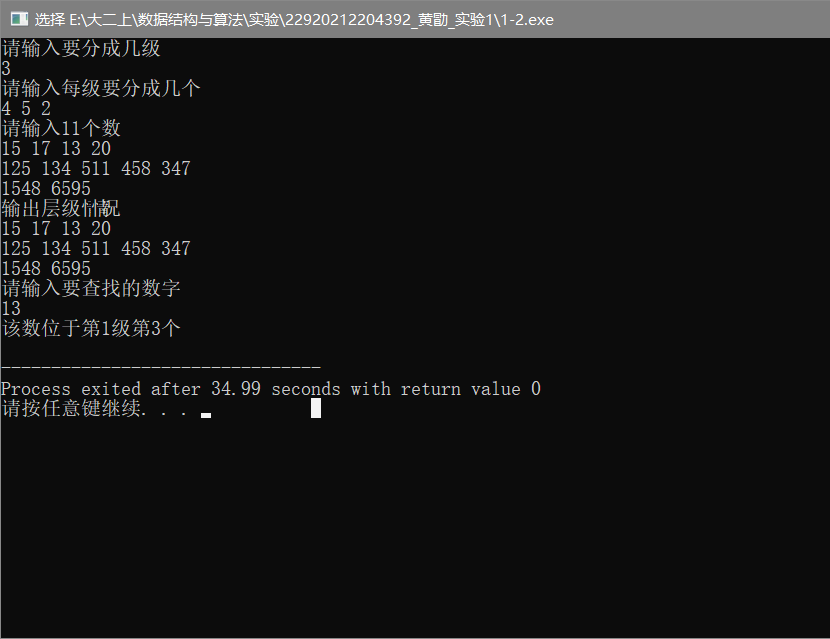
（结合截图说明算法的输入输出）

1. 关于1-1的输入与输出：



在实际运行中，我创建了1->3->5->7->9的链表，在程序中能够得到良好展示，并且实现了查找结点和删除节点的功能。

1. 关于1-2的输入与输出



在实际运行中，我创建了15 17 13 20 125 134 511 458 347 1548 6595的顺序表，并且分成了三级，通过编写的查找算法我成功找到了13位于第1级第3个，程序整个过程运行良好。

1. **实验小结（即总结本次实验所得到的经验与启发等）：**

在本次实验中，我尝试具体运用顺序表以及链表，在实体机的实验中我能够更深刻地理解对这一部分数据结构的执行方式与特点，并且在编写代码的过程中，我通过不断的调试去寻找语句之间的问题和不足，在潜移默化中提高了我的代码编写能力，这是一次完成效果良好的实验！