

COMPUTER

高等院校计算机技术



“十二五”规划教材

数据结构实验教程

◎ 严 冰 柳 俊 张 泳 编著
王云武 胡 隽 王泽兵



ZHEJIANG UNIVERSITY PRESS
浙江大学出版社

高等院校实验教材

数据结构实验教程

严 冰 柳 俊 张 泳 王云武 胡 隽 王泽兵 编著



ZHEJIANG UNIVERSITY PRESS

浙江大学出版社

内容简介

本书是作者积多年讲授“数据结构”课程及指导学生实验的教学经验而编写的一本实验教材。全书采用 C++ 语言作为数据结构与算法的描述语言,对应于教科书中的各知识点,每一章首先对知识点进行概述,然后给出相应内容的若干个实验项目,最后再给出习题范例解析与习题以巩固知识点的掌握。

本书由 2 个篇章组成,第一篇章是数据结构的基础部分,内容涉及数据结构和算法分析基础、线性表、栈和队列、树和二叉树、图;第二篇章是数据结构的进阶部分,内容涉及线性表和栈的应用、稀疏矩阵和广义表、特殊二叉树、图的应用、查找与排序等。每个知识点均包含 2 至 3 个实验项目,实验内容的组织充分顾及了不同的难易程度,每个实验项目除给出基本实验内容外,还包含选做内容部分与实验提示,以符合不同层次的学生。此外,每个知识点还给出了习题范例解析、选择题、填空题、解答题、算法设计题及参考答案,这些题目大多是作者长年教学的积累。

本书实验内容丰富、习题题型多样、实用性强,非常适合作为应用型高等院校计算机及相关专业数据结构课程的实验教材,也可供各类学生课程学习与考前复习使用。

图书在版编目 (CIP)数据

数据结构实验教程 / 严冰等编著. —杭州: 浙江大学出版社, 2012.7
ISBN 978-7-308-10008-3

I. ①数… II. ①严… III. ①数据结构—高等学校—教材 IV. ①TP311.12

中国版本图书馆 CIP 数据核字 (2012) 第 101923 号

数据结构实验教程

严 冰 柳 俊 张 泳 王云武 胡 隽 王泽兵 编著

责任编辑 周卫群

封面设计 刘依群

出版发行 浙江大学出版社

(杭州市天目山路 148 号 邮政编码 310007)

(网址: <http://www.zjupress.com>)

排 版 杭州中大图文设计有限公司

印 刷 富阳市育才印刷有限公司

开 本 787mm×1092mm 1/16

印 张 14.75

字 数 377 千

版 次 2012 年 7 月第 1 版 2012 年 7 月第 1 次印刷

书 号 ISBN 978-7-308-10008-3

定 价 30.00 元

版权所有 翻印必究 印装差错 负责调换

浙江大学出版社发行部邮购电话 (0571)88925591

前 言

数据结构是计算机及相关专业的一门重要的专业基础课,它所讨论的知识内容和提倡的技术方法,无论对于进一步学习计算机领域的其他课程,还是对从事软件工程的开发,都有着不可替代的作用。而高等教育的大众化对数据结构课程的教学提出了新的要求,新的高等教育形势需要我们积极研究新的教学方法。在长期的教学实践中我们体会到,“因材施教”,把实践环节与理论教学相融合,通过实践教学促进学科理论知识的学习,是有效地提高教学效果和教学水平的重要方法之一。为完善数据结构课程的实验教学,为社会培养合格和适用的专业人才,我们依据多年来讲授“数据结构”课程及指导学生实验的教学经验,在实验内容选择和安排等方面做了精心考虑,编写了这本实验教材。

本书编写原则是:依据课程教学大纲,充分理解课程的大多数主教材,遵循教学的规律和节奏,充分体现实验的可操作性和可选择性,可以与课程主教材辅助配套使用。另一方面,由于本书每一章除了内容丰富的实验项目外,还包含了必要的理论知识概述及习题解析,所以,还可以作为良好的自学教材,供各类学生课程学习与考前复习使用。

本书采用 C++ 语言作为数据结构与算法的描述语言,对应于教科书中的各知识点,每一章首先对知识点进行概述,然后给出相应内容的若干个实验项目,最后再给出习题范例解析与习题。全书由 2 个篇章组成,第一篇章是数据结构的基础部分,内容涉及数据结构和算法分析基础、线性表、栈和队列、树和二叉树、图;第二篇章是数据结构的进阶部分,内容涉及线性表和栈的应用、稀疏矩阵和广义表、特殊二叉树、图的应用、查找与排序等。每个知识点均包含 2 至 3 个实验项目,实验内容的组织充分顾及了不同的难易程度,每个实验项目除给出基本实验内容外,还包含选做内容部分与实验提示,以符合不同层次的学生。此外,每一章还给出了习题范例解析、选择题、填空题、解答题、算法设计题,以及所有习题的参考答案,这些题目大多是作者长年教学积累的成果,通过习题,希望帮助读者加深对每一个知识点的理解和掌握。

本书由浙江大学城市学院长期担任“数据结构”课程的教师集体编写,编著者们承担的“数据结构”课程是杭州市精品课程。该书中,严冰设计了整体框架,并编写了第二章、第六章、第七章及附录;柳俊编写了第四章和第八章;张泳编写了第十章和第十一章;王云武编写了第一章和第三章;胡隽编写了第五章和第九章。全书由严冰负责统稿,课程建设负责人王泽兵教授参与了该书的整体思路设计。本书的编写得到了浙江大学城市学院刘加海教授的大力支持,在此深表感谢!

本书虽经多次修改,仍可能存在不妥乃至错误之处,敬请各位专家和广大读者不吝赐教!

编 者

2012 年 1 月

第一篇 数据结构基础

第 1 章 数据结构与算法分析基础	3
1.1 知识点概述	3
1.1.1 基本术语	3
1.1.2 算法和算法的量度	4
1.1.3 C++ 相关知识	4
1.2 实验项目	5
1.2.1 熟悉 Project 组织应用程序实验	5
1.2.2 抽象数据类型的表示和实现实验	8
1.2.3 算法和算法分析实验	11
1.3 习题范例解析	14
1.4 习题	17
1.4.1 选择题	17
1.4.2 填空题	18
1.4.3 应用题	20
第 2 章 线性表	22
2.1 知识点概述	22
2.1.1 线性表的定义和抽象数据类型	22
2.1.2 线性表的顺序存储表示	23
2.1.3 线性表的链接存储表示	24
2.2 实验项目	26
2.2.1 线性表的顺序存储实验	26
2.2.2 线性表的链式存储实验	28
2.2.3 约瑟夫环的实现	29
2.3 习题范例解析	30
2.4 习题	32
2.4.1 选择题	32
2.4.2 填空题	34
2.4.3 应用题	35
2.4.4 算法设计题	36

第3章 栈和队列	45
3.1 知识点概述	45
3.1.1 栈	45
3.1.2 队列	47
3.2 实验项目	50
3.2.1 栈的顺序表示和实现实验	50
3.2.2 队列(循环队列)的表示和实现实验	51
3.2.3 栈与队列的应用实验	52
3.3 习题范例解析	53
3.4 习题	59
3.4.1 选择题	59
3.4.2 填空题	62
3.4.3 应用题	63
3.4.4 算法设计题	64
第4章 树	75
4.1 知识点概述	75
4.1.1 树的定义和基本概念	75
4.1.2 二叉树的定义和基本概念	76
4.1.3 二叉树的存储结构	78
4.1.4 二叉树的遍历	79
4.2 实验项目	80
4.2.1 二叉链表的基本操作实验	80
4.2.2 二叉链表的进一步操作实验	81
4.3 习题范例解析	83
4.4 习题	85
4.4.1 选择题	85
4.4.2 填空题	86
4.4.3 应用题	87
4.4.4 算法设计题	88
第5章 图	92
5.1 知识点概述	92
5.1.1 图的定义	92
5.1.2 图的基本术语	92
5.1.3 图的抽象数据类型	94
5.1.4 图的存储结构	94
5.1.5 图的遍历	97
5.2 实验项目	98
5.2.1 图的基本操作实验——邻接矩阵存储结构	98

5.2.2 图的基本操作实验——邻接表存储结构	99
5.3 习题范例解析	101
5.4 习题	102
5.4.1 选择题	102
5.4.2 填空题	103
5.4.3 应用题	104
5.4.4 算法设计题	105

第二篇 数据结构进阶

第6章 线性表和栈的应用	111
6.1 知识点概述	111
6.1.1 线性表的应用——多项式计算	111
6.1.2 栈的应用——算术表达式的计算	112
6.2 实验项目	114
6.2.1 线性表的应用——多项式计算实验	114
6.2.2 栈的应用——算术表达式的计算实验	116
6.3 习题范例解析	117
6.4 习题	118
6.4.1 选择题	118
6.4.2 填空题	119
6.4.3 应用题	120
6.4.4 算法设计题	121
第7章 稀疏矩阵和广义表	126
7.1 知识点概述	126
7.1.1 稀疏矩阵	126
7.1.2 广义表	129
7.2 实验项目	130
7.2.1 稀疏矩阵的顺序存储实验	130
7.2.2 稀疏矩阵的链接存储实验	131
7.2.3 广义表运算实验	133
7.3 习题范例解析	134
7.4 习题	135
7.4.1 选择题	135
7.4.2 填空题	136
7.4.3 应用题	136
7.4.4 算法设计题	137
第8章 特殊二叉树	142
8.1 知识点概述	142

8.1.1	二叉搜索树	142
8.1.2	堆	145
8.1.3	哈夫曼树	147
8.2	实验项目	149
8.2.1	二叉搜索树的基本操作实现	149
8.2.2	堆的基本操作实现	150
8.2.3	哈夫曼树及其应用实验	151
8.3	习题范例解析	152
8.4	习题	153
8.4.1	选择题	153
8.4.2	应用题	155
8.4.3	算法设计题	160
第9章	图的应用	163
9.1	知识点概述	163
9.1.1	最小生成树	163
9.1.2	最短路径	165
9.1.3	拓扑排序	167
9.2	实验项目	168
9.2.1	图的最小生成树实验	168
9.2.2	图的最短路径实验	170
9.2.3	图的拓扑排序实验	173
9.3	习题范例解析	175
9.4	习题	179
9.4.1	选择题	179
9.4.2	填空题	180
9.4.3	应用题	181
9.4.4	算法设计题	183
第10章	查找	188
10.1	知识点概述	188
10.1.1	查找的概念	188
10.1.2	顺序查找	188
10.1.3	二分查找	189
10.1.4	索引查找	189
10.1.5	散列查找	190
10.2	实验项目	193
10.2.1	索引查找的实现	193
10.2.2	散列查找的实现	195
10.3	习题范例解析	197

10.4 习题	199
10.4.1 选择题	199
10.4.2 填空题	201
10.4.3 应用题	201
10.4.4 算法设计题	202
第 11 章 排序	206
11.1 知识点概述	206
11.1.1 排序的基本概念	206
11.1.2 插入排序	206
11.1.3 选择排序	207
11.1.4 交换排序	208
11.1.5 归并排序	209
11.1.6 内排序方法比较	210
11.2 实验项目	210
11.2.1 排序算法的应用(一)	210
11.2.2 排序算法的应用(二)	212
11.3 习题范例解析	213
11.4 习题	215
11.4.1 选择题	215
11.4.2 填空题	217
11.4.3 应用题	218
11.4.4 算法设计题	219
附录 实验报告格式	223
主要参考文献	224

第一篇

数据结构基础

第 1 章 数据结构与算法分析基础

1.1 知识点概述

数据结构是一门讨论“描述现实世界实体的数学模型(非数值计算)及其上的操作在计算机中如何表示和实现”的学科。

1.1.1 基本术语

数据是计算机操作对象的总称,它是计算机处理的符号的集合,集合中的个体为一个数据元素。

数据元素可以是不可分割的原子,也可以由若干数据项合成,因此在数据结构中讨论的基本单位是数据元素,而最小单位是数据项。

数据结构是由若干特性相同的数据元素构成的集合,且在集合上存在一种或多种关系。根据关系不同可将数据结构分为四类(称为数据的逻辑结构),如图 1.1 所示。

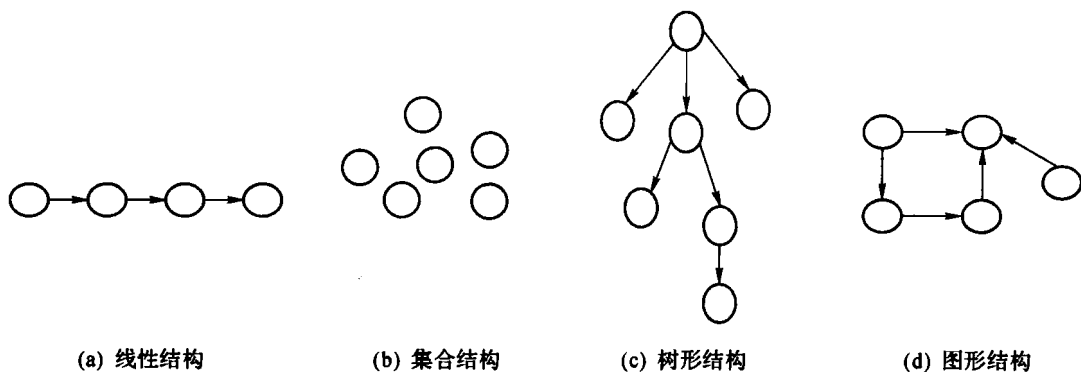


图 1.1 四类常见的数据结构

数据的存储结构是数据逻辑结构在计算机中的映像,由关系的两种映像方法可得到两类存储结构:一类是顺序存储结构,它以数据元素相对的存储位置表示关系,则存储结构中只包含数据元素本身的信息;另一类是链式存储结构,它不仅仅包含数据元素本身的信息,并附加的指针信息(后继元素的存储地址)表示关系。

数据结构的操作是和数据结构本身密不可分的,两者作为一个整体可用抽象数据类型进行描述。抽象数据类型是一个数学模型以及定义在该模型上的一组操作,因此它和高级程序设计语言中的数据类型具有相同含义,而抽象数据类型的范畴更广,它不局限于现有程序设计语言中已经实现的数据类型,但抽象数据类型需要借用固有数据类型表示并实现。抽象数据

类型的三大要素为数据对象、数据关系和基本操作,同时数据抽象和数据封装是抽象数据类型的两个重要特性。

1.1.2 算法和算法的量度

算法是进行程序设计的不可缺少的要素。算法是对问题求解的一种描述,是为解决一个或一类问题给出的一种确定规则的描述。一个完整的算法应该具有下列五个要素:有穷性、确定性、可行性、有输入和有输出。一个正确的算法应对苛刻且带有刁难性的输入数据也能得出正确的结果,并且对不正确的输入也能做出正确的反映。算法设计的原则主要包含以下几个方面:正确性、可读性、健壮性以及高效率与低存储量需求。

算法的时间复杂度是比较不同算法效率的一种准则,算法时间复杂度的估算基于算法中基本操作的重复执行次数,或处于最深层循环内的语句的频度。常见的时间复杂度,按数量级递增排列依次为:常数阶 $O(1)$ 、对数阶 $O(\log_2 n)$ 、线性阶 $O(n)$ 、线性对数阶 $O(n \log_2 n)$ 、平方阶 $O(n^2)$ 、立方阶 $O(n^3)$ 、指数阶 $O(2^n)$ 、阶层阶 ($O(n!)$)、指数阶 ($O(n^n)$)。

算法空间复杂度可作为算法所需存储量的一种量度,它主要取决于算法的输入量和辅助变量所占空间,若算法的输入仅取决于问题本身而和算法无关,则算法空间复杂度的估算只需考察算法中所用辅助变量所占空间,若算法的空间复杂度为常量级,则称该算法为原地工作的算法。

1.1.3 C++ 相关知识

1. 自定义类型

为了使程序便于理解,应该把一些数据类型名与要说明的对象性质建立某些语义上的对应关系,可以用 typedef 向用户提供了一种自定义类型说明符,一般形式为:

typedef 类型 定义名;

使用 typedef 定义数据类型新名字既照顾了用户编程使用词汇的习惯,又增加了程序的可读性,便于程序的移植。用 typedef 定义类型,只定义了一个数据类型的新名字而不是定义一种新的数据类型。

2. 输入输出语句

使用包含文件语句 `#include <iostream.h>` 后,可使用以下关键字进行输入/输出:

`cin` 代表标准输入设备(键盘)流对象,后面跟“>>”运算符,“>>”是 C++ 的输入运算符;

`cout` 代表标准输出设备(屏幕)流对象,后面跟“<<”运算符,“<<”是 C++ 的输出运算符;

`cerr` 代表标准错误输出设备(屏幕)流对象,后面跟“<<”运算符。

3. 引用参数

在 C 语言中,可以使用指针参数达到修改形参所指向的变量的值,但指针的间接性给人一种不实在感。C++ 引入引用的主要目的是建立某种类型的虚实体,这种虚实体不占有实际的存储空间。它作为函数参数时,从实参得到相应的地址,与实参共用相同的存储空间,也就是被调用函数中形参的变化会导致所对应实参的变化。

使用方法是在形式参数之前插入“&”符号。

4. 动态存储分配

C++的 new 运算符和 delete 运算符提供了动态分配内存的功能。new 的功能是给程序实体动态地分配存储空间,delete 运算符的功能是将用 new 运算符动态分配的空间回收。

new 和 delete 运算符都是单目运算符,new 运算符不能对动态分配的存储区进行初始化。

1.2 实验项目

1.2.1 熟悉 Project 组织应用程序实验

1. 实验目的

- (1) 熟悉 VC6.0 开发环境;使用 Workspace 和 Project 组织应用程序;
- (2) 回顾 C 语言程序设计,编写完整的实验应用程序,并调试通过。
- (3) 掌握文件包含,以及库函数 `iostream.h` 中的标准输入、输出流对象 `cin` 和 `cout`,理解“引用(&)”方式的参数传递。

2. 实验内容

- (1) 按下述介绍的方法,练习并掌握使用 Project 组织应用程序的方法。

步骤:

① 启动 VC++,选择“文件(File)”菜单中的“新建(New)”项,选择“工程(Projects)页”,然后选择“Win32 Console Application”,在右上角输入 project 的名称(如:Example),再选择合适的存储路径,然后按下“确定”按钮。这样就建立了一个新的工程。如图 1.2 所示。

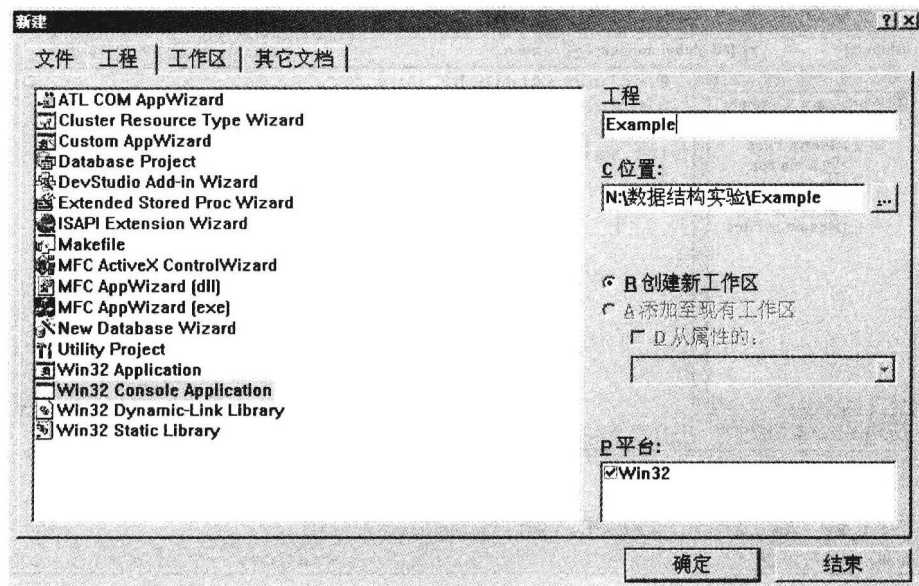


图 1.2 新建工程对话框

② 在窗口左侧出现 WorkSpace 视图,选择“FileView”页可浏览该工程所包含的文件。然后可在项目中新建源文件(菜单:文件→新建),包括“C/C++ Header File”和“C/C++ Source File”两类文件,或将已有的源文件加入到这个工程中(菜单:工程→增加到工程)。如图 1.3 所示。

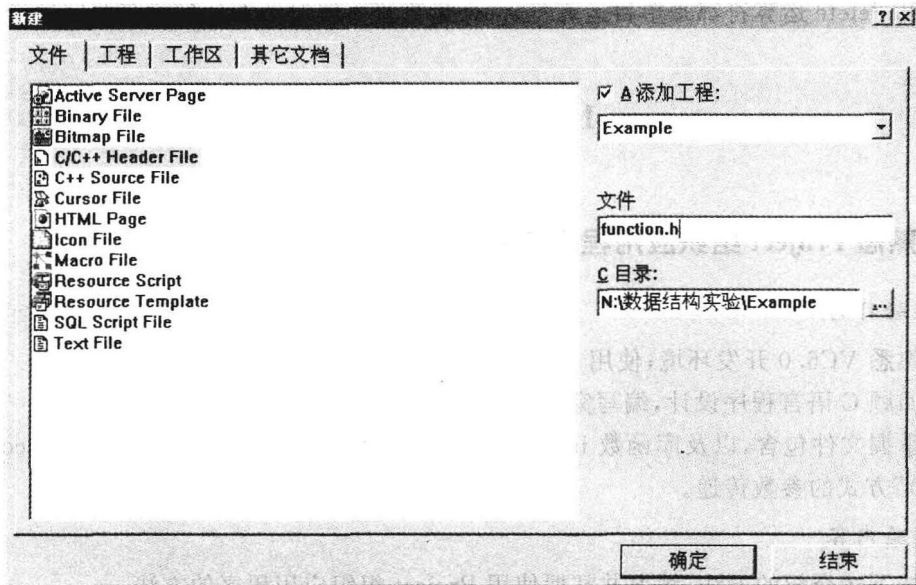


图 1.3 新建文件对话框

完成后程序组织结构如图 1.4 所示。

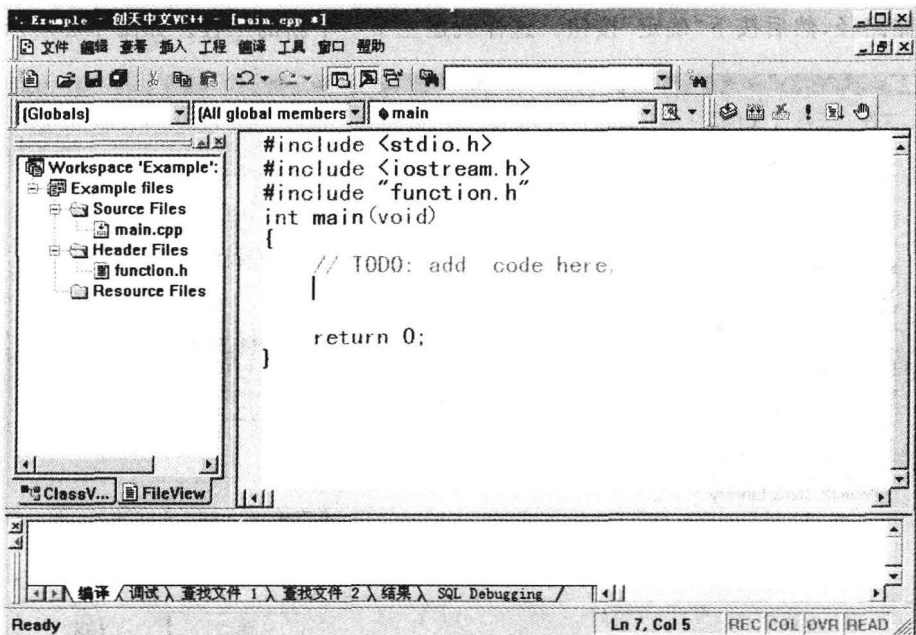


图 1.4 工程结构示意图

其中 Source Files 中包含主程序等源程序文件(如:main.cpp),Header Files 中包含头文

件等(如: function.h)。

(2) 在 VC 中建立工程(工程名为 test1_1), 添加头文件(test1_1.h)和源文件(test1_1.cpp)并编写如下程序加入到工程中, 编译执行该程序。要求使用 cin 和 cout 进行数据的输入输出。

程序要求如下:

设 a 为长度为 n 的整数型一维数组。

①编写求 a 中的最大值、最小值和平均值的函数 void aMAX_MIN_AVE(int *a, int n, int &max, int &min, int &aver), 用“引用参数”带回结果。

②编写函数 int prime_SUM(int *a, int n) 计算 a 中所有素数之和。

③选做: 编写函数 void aSORT(int *a, int n) 对 a 进行从小到大的排序, 并输出排序结果。

(3) 填写实验报告。

3. 实验提示

(1) test1_1.h 文件框架:

```
void aMAX_MIN_AVE(int *a, int n, int &max, int &min, int &aver)
{ //求数组 a 中的最大值、最小值和平均值
    .....
}

int prime_SUM(int *a, int n)
{ //计算数组 a 中元素值是素数的各元素之和
    .....
}

void aSORT(int *a, int n)
{ // 对数组 a 进行排序
    .....
}
```

(2) test1_1.cpp 文件框架:

```
#include <stdio.h>
#include <iostream.h>
#include "test1_1.h"
int main(void)
{
    int a[10], x, aMax, aMin, aVer;
    int i = 0;
    cin >> x;
    while(x != -1 && i < 10)
    {
        a[i++] = x;
        cin >> x;
    }
    cout << "数组中的元素为:" << endl;
```



```

for(j = 0; j < i; j++)
    cout << a[j] << " ";
cout << endl;

.....// 调用函数 aMAX_MIN_AVE

cout << "最大值 = " << aMax << endl;
cout << "最小值 = " << aMin << endl;
cout << "平均值 = " << aVer << endl;
cout << "数组中所有素数和 = " << .....;    // 调用函数 prime_SUM

..... // 调用函数 aSORT

cout << "排序后数组元素为:" << endl;
for(j = 0; j < i; j++)
    cout << a[j] << " ";
cout << endl;
return 0;
}

```

1.2.2 抽象数据类型的表示和实现实验

1. 实验目的

- (1) 通过抽象数据类型三元组的表示和实现,了解抽象数据类型的定义方式。
- (2) 掌握抽象数据类型的定义方式和用 C 语言实现的方法。
- (3) 熟悉如何运用主函数检验各基本操作函数的正确性的具体操作

2. 实验内容

- (1) 认真阅读以下有关抽象数据类型的知识:

① 抽象数据类型的概念

抽象数据类型是指一个数学模型以及定义在该模型上的一组操作。抽象数据类型的定义仅取决于它的一组逻辑特性,而与其在计算机内部如何表示和实现无关,即不论其内部结构如何变化,只要它的数学特性不变,就不影响其外部的使用。

一个含抽象数据类型的软件模块通常应包含定义、表示和实现 3 个部分。抽象数据类型通常采用以下格式定义:

```

ADT 抽象数据类型名 {
    数据对象:<数据对象的定义>
    数据关系:<数据关系的定义>
    基本操作:<基本操作的定义>
} ADT 抽象数据类型名

```

② 三元组的抽象数据类型定义及表示

我们以抽象数据类型三元组为例,说明抽象数据类型是如何定义的。三元组实际上就是一个数据对象中有 3 个数据元素。三元组中元素的数据类型,可以是整型数,也可以是字符、