

可下载教学资料

<http://www.tup.tsinghua.edu.cn>

21世纪普通高校计算机公共课程规划教材

数据结构实验 与实训教程 (第3版)

邓文华 主编 胡智文 副主编

清华大学出版社

21 世纪普通高校计算机公共课程规划教材

数据结构实验 与实训教程

(第3版)

邓文华 主编 胡智文 副主编

清华大学出版社
北京

内 容 简 介

本书是用于“数据结构”课程的辅助教材。根据教学内容及针对独立学院三本学生的实际情况,本书在内容编排上共分成4部分:第一部分为概述,包括C语言的数据输入、结构体概念、函数的传址调用概念及预备知识实验;第二部分为基础实验部分,给出了11个实验,包括线性结构、树形结构、图形结构、查找、排序以及数组和字符串等操作;第三部分为课程设计部分,包括航空客运订票系统、汉诺塔游戏程序、全屏编辑程序设计、旅游路线安排模拟系统、停车场管理及最小生成树 Kruskal 6个综合性实验,此部分实验可作为数据结构的课程设计;第四部分为满足教学和各类学生学习课程与考前复习的需要,安排了12套模拟试题,并给出了详细的解答。

本书内容丰富、概念清晰、实用性强。它与《数据结构》的主要内容紧密结合,可供普通高等院校计算机及相关专业学生学习、实验、课程设计与考前复习使用,也可供教师或其他专业技术人员参考。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

数据结构实验与实训教程/邓文华主编.--3版.--北京:清华大学出版社,2011.1

(21世纪普通高校计算机公共课程规划教材)

ISBN 978-7-302-23640-5

I. ①数… II. ①邓… 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(2010)第160123号

责任编辑:梁颖 顾冰

责任校对:白蕾

责任印制:杨艳

出版发行:清华大学出版社

地 址:北京清华大学学研大厦A座

<http://www.tup.com.cn>

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62795954, jsjic@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者:北京市清华园胶印厂

经 销:全国新华书店

开 本:185×260 印 张:12.25 字 数:289千字

版 次:2011年1月第3版 印 次:2011年1月第1次印刷

印 数:1~3000

定 价:22.00元

产品编号:037211-01

出版说明

随着我国改革开放的进一步深化,高等教育也得到了快速发展。各地高校紧密结合地方经济建设发展需要,科学运用市场调节机制,加大了使用信息科学等现代科学技术提升、改造传统学科专业的投入力度。通过教育改革合理调整和配置了教育资源,优化了传统学科专业,积极为地方经济建设输送人才,为我国经济社会的快速、健康和可持续发展以及高等教育自身的改革发展做出了巨大贡献。但是,高等教育质量还需要进一步提高以适应经济社会发展的需要,不少高校的专业设置和结构不尽合理,教师队伍整体素质亟待提高,人才培养模式、教学内容和教学方法需要进一步转变,学生的实践能力和创新精神亟待加强。

教育部一直十分重视高等教育质量工作。2007年1月,教育部下发了《关于实施高等学校本科教学质量与教学改革工程的意见》,计划实施“高等学校本科教学质量与教学改革工程(简称‘质量工程’)”,通过专业结构调整、课程教材建设、实践教学改革、教学团队建设等多项内容,进一步深化高等学校教学改革,提高人才培养的能力和水平,更好地满足经济社会发展对高素质人才的需要。在贯彻和落实教育部“质量工程”的过程中,各地高校发挥师资力量强、办学经验丰富、教学资源充裕等优势,对其特色专业及特色课程(群)加以规划、整理和总结,更新教学内容,改革课程体系,建设了一大批内容新、体系新、方法新、手段新的特色课程。在此基础上,经教育部相关教学指导委员会专家的指导和建议,清华大学出版社在多个领域精选各高校的特色课程,分别规划出版系列教材,以配合“质量工程”的实施,满足各高校教学质量和教学改革的需要。

本系列教材立足于计算机公共课程领域,以公共基础课为主、专业基础课为辅,横向满足高校多层次教学的需要。在规划过程中体现了如下一些基本原则和特点。

(1) 面向多层次、多学科专业,强调计算机在各专业中的应用。教材内容坚持基本理论适度,反映各层次对基本理论和原理的需求,同时加强实践和应用环节。

(2) 反映教学需要,促进教学发展。教材要适应多样化的教学需要,正确把握教学内容和课程体系的改革方向,在选择教材内容和编写体系时注意体现素质教育、创新能力与实践能力的培养,为学生知识、能力、素质协调发展创造条件。

(3) 实施精品战略,突出重点,保证质量。规划教材把重点放在公共基础课和专业基础课的教材建设上;特别注意选择并安排一部分原来基础比较好的优秀教材或讲义修订再版,逐步形成精品教材;提倡并鼓励编写体现教学质量和教学改革成果的教材。

(4) 主张一纲多本,合理配套。基础课和专业基础课教材配套,同一门课程针对不同层次、面向不同专业的多本具有各自内容特点的教材。处理好教材统一性与多样化,基本教材与辅助教材、教学参考书,文字教材与软件教材的关系,实现教材系列资源配套。

(5) 依靠专家,择优选用。在制定教材规划时要依靠各课程专家在调查研究本课程教

材建设现状的基础上提出规划选题。在落实主编人选时,要引入竞争机制,通过申报、评审确定主题。书稿完成后要认真实行审稿程序,确保出书质量。

繁荣教材出版事业,提高教材质量的关键是教师。建立一支高水平教材编写梯队才能保证教材的编写质量和建设力度,希望有志于教材建设的教师能够加入到我们的编写队伍中来。

21 世纪普通高校计算机公共课程规划教材编委会

联系人:梁颖 liangying@tup.tsinghua.edu.cn

前 言

本书为独立学院三本《数据结构》一书的配套教材,结合独立学院三本的特点,在原来第2版的基础上进行了修改与补充。其任务是通过实践,让学生进一步掌握常用数据结构的基本概念及其不同的实现方法,并对在不同存储结构上实现不同的运算方法和技巧有所体会。本教材是专门针对独立学院三本学生的实际情况,为学习“数据结构”课程而编写的实验教材。本书共分成4部分:第一部分为概述,包括C语言的数据输入、结构体概念、函数的传址调用概念及预备知识实验;第二部分为基础实验部分,在内容安排上给出了11个实验,包括线性结构、树形结构(二叉树的二叉链表存储方式、结点结构、类型定义、基本运算及应用)、图形结构(图的两种存储结构的表示方法)、查找(顺序查找、树表查找、散列表查找的基本思想及存储、运算的实现)、排序(插入排序、冒泡排序、快速排序、直接选择排序、堆排序、归并排序及基数排序的基本思想及实现)以及数组和字符串的操作;第三部分为课程设计部分,包括航空客运订票系统、汉诺塔游戏程序、全屏幕编辑程序设计、旅游路线安排模拟系统、停车场管理及最小生成树 Kruskal 6个综合性实验,这组实验的综合性比较强,此部分可作为《数据结构》课程的课程设计之用;第四部分安排了12套模拟试题及其参考解答,目的是为了帮助学生进一步巩固所学的理论知识。

本教材具有以下几个特点:

(1) 每个实验题目都给出相应的C程序模板,在模板中填写关键语句或子程序即可上机运行,便于学生集中精力于主要的算法。

(2) 实验内容安排多样性。包括基础题和提高题,便于满足不同层次学生的需求。形式包括:给出程序框架要求填写关键算法的形式;给出类似函数要求独立编写程序的形式;给出主程序要求编写子程序的形式以及给出算法要求编写程序等形式。

(3) 第四部分安排了12套模拟试题,并给出详细的参考解答,有利学生的自学、复习。

本书由邓文华任主编,胡智文任副主编。其中第二、三部分由邓文华修改、编写;第四部分由胡智文修改、编写;第一部分由毕保祥编写。全书由邓文华统稿。

本书中的所有程序都在 Borland C++ 软件开发环境下调试运行通过。

编写独立学院三本的教材是一项新的尝试,难免存在疏漏,恳请读者赐教指正。

编者

2010年8月

目 录

第一部分 概述	1
预备知识	1
预备知识实验	7
预备知识实验参考答案	12
第二部分 基础实验	13
实验 1 线性表的基本操作	13
实验 2 链表的基本操作	18
实验 3 栈的基本操作	23
实验 4 队列的基本操作	29
实验 5 数组的基本操作	39
实验 6 字符串的基本操作	42
实验 7 二叉树的基本操作	47
实验 8 树的遍历和哈夫曼树	52
实验 9 图的基本操作	61
实验 10 排序	67
实验 11 查找	72
基础实验参考答案	76
第三部分 课程设计实验	88
实验 1 航空客运订票系统	88
实验 2 汉诺塔游戏程序	93
实验 3 全屏幕编辑程序设计	98
实验 4 旅游路线安排模拟系统	107
实验 5 停车场管理	111
实验 6 最小生成树 Kruskal 算法	111
第四部分 模拟试题	116
模拟试题 1	116
模拟试题 2	118

模拟试题 3	122
模拟试题 4	125
模拟试题 5	128
模拟试题 6	130
模拟试题 7	133
模拟试题 8	137
模拟试题 9	140
模拟试题 10	143
模拟试题 11	146
模拟试题 12	150
模拟试题参考答案	154
附录	179
附录 A 实验报告参考规范	179
附录 B Borland C++ 上机基本操作	180
参考文献	183

预 备 知 识

数据结构不仅具有较强的理论性,更具有较强的实践性。在数据结构教学过程中,如果学生的程序设计语言基础薄弱,就会在很大程度上影响正常的教学进度。因此,为了方便学生的后续实验,这里将数据结构所必需的 C 语言语法做简单介绍。根据多年教学实践,学生完成上机实验时遇到的主要问题包括:不能正确地输入数据,不熟悉结构体概念,函数的传址调用概念不清,指针与链表比较生疏。由于篇幅所限,这里仅对前三个问题加以介绍,关于指针与链表知识请参阅相应的 C 语言程序设计教程。

1. 基本输入和输出

上机实践离不开数据的输入和输出。对于一个算法程序,如果数据不能正确输入,算法设计得再好也无法正常运行。虽然输入/输出看起来简单,但是往往在上机实验时最容易出错,尤其是输入。

1) scanf()输入函数

C 语言的输入由系统提供的函数实现,如 scanf()函数等。因为标准输入/输出函数都在于头文件 stdio.h 之中声明,所以只有在程序中将其包含进来方可使用。在程序的首部一般要求写入:

```
#include <stdio.h>
```

函数 scanf()的功能很丰富,输入格式也是多种多样,这里只给出在使用时需要注意几个问题。

(1) 当一条 scanf()语句中有多个数值型变量(如 int、float、double)时,在输入数据时可在一行之内输入多个数据,数据之间空格分隔;如果语句中在 %d 和 %f 之间有一个逗号“,”,就要在数据之间以“,”分隔。例如:

```
int x; float y;  
scanf ("%d %f", &x, &y);
```

正确的输入应是:

整数 空格 实数 回车

例如:

50 78.3

就是在两个数据之间使用空格键作为分隔符,最后按 Enter 键。例如:

```
scanf("%d", %f", &x, &y);
```

在 %d 和 %f 之间有一个逗号,正确的输入方法是:

整数 逗号 实数 回车

例如:

50,78.3

(2) 不要将字符型变量或字符串与数值型变量混合在一条 scanf() 语句中,应该各自单独写一条输入语句,这样不易出错。“空格键”在数值型数据之间起“分隔符”作用,而在字符或字符串之间,“空格”则被当做一个输入的字符,不能起到“分隔符”的作用。

(3) scanf() 函数中必须传递变量的地址。因为数组名本身就是数组的首地址,所以在 scanf 语句中字符数组名字之前不得冠以取地址符“&”,而整型变量、实型变量和字符型变量之前必须加上“&”。

2) printf() 输出函数

C 语言的输出是由系统提供的 printf() 等函数来实现的,与使用 scanf() 函数一样,在程序的首部也要求写入:

```
#include <stdio.h>
```

关于 printf() 函数,这里只强调如下几点:

(1) 在输入语句 scanf() 之前,应该先使用 printf() 语句进行必要的提示。例如:

```
char name[10], x;  
int y;  
printf("\n 请输入学生的姓名、性别和成绩:");  
scanf("%s", name);  
scanf("%c", &x);  
scanf("%d", &y);
```

(2) 在该换行的地方,要及时换行。例如:

```
int i;  
printf("数据输出如下: \n"); /* 输出并换行 */  
for (i = 1; i <= 15; i++) {  
    printf("%5d", i * i);  
    if(i % 5 == 0) printf("\n"); /* 每行输入 5 个数据 */  
}
```

(3) 在调试程序时适当添加一些输出语句,以便监视程序运行状态。程序调试成功后,再将这些输出语句去掉。

2. 结构体的运用

在 C 语言中结构体的定义、输入/输出语句是数据结构程序设计的重要语法基础。

1) 定义结构体类型的一般格式

```
struct 结构体类型名
```

```
{  类型名 1      成员名 1;
    类型名 2      成员名 2;
    :
    类型名 n      成员名 n;
};
```

其中,每个成员都是结构体的一个数据子域。数据子域的类型名一般指基本数据类型(int char 等),也可以是数组,还可以是已经定义的另一结构体名。

2) 定义结构体变量的方法

struct 结构体类型名 结构体变量名 1, 结构体变量名 2, ..., 结构体变量名 m;

例如:

```
struct  ElemType    /* 定义结构体类型 */
{ int  num;
  char name[10];
};
struct  ElemType  x; /* 定义结构体变量 x */
```

3) 用 typedef 语句为结构体类型定义新的类型名

例如:

```
typedef  struct
{ int  num;
  char name[10];
} ElemType;
```

其中,ElemType 就是结构体类型的一个新类型名。这样,定义变量 x 的语句就可以写为

```
ElemType  x;
```

4) 结构体变量的引用

(1) 用“.”引用:

<结构体变量名>.<成员名>

例如:

```
x.num = 10;
scanf("% s", x.name);
```

使用方法与同类型的简单变量一样。

(2) 用“->”引用:

<指针变量名> -> <成员名>

其中,<指针变量名>必须是指向同类型结构体的指针变量。例如:

```
ElemType  * pt;
pt = &x;          /* pt 保存 x 的地址,常称为使 pt 指向变量 x */
```

形如 `pt->num` 和 `pt->name` 的表示读作“`pt` 所指的 `num`”和“`pt` 所指的 `name`”。使用方法与同类型的简单变量一样。

5) 一个简单的结构体运用程序

设一个人的记录如下:

姓名: Zhang San

年龄: 20

性别: 男

身高: 175 厘米

体重: 68 千克

下面用 `struct` 定义这位男生的记录结构,并用一个结构体变量保存其记录的值。另外,用两个变量分别演示用“.”和“->”方法引用结构体的成员。

```
main(){
    struct person {
        char name[20];
        int age;
        char sex;
        int height;
        int weight;
    } stu1, *p;
    /* 初始化 stu3 */
    struct person stu3 = {"Zhang San", 20, 'M', 175, 68};
    printf("    %s %d %c %d %d\n", stu3.name, stu3.age = 20, stu3.sex, stu3.height,
stu3.weight);
    /* 初始化 stu1 */
    strcpy(stu1.name, "Zhang San");
    stu1.age = 20;  stu1.sex = 'M';
    stu1.height = 175;  stu1.weight = 68;
    printf("    %s %d %c %d %d\n", stu1.name, stu1.age = 20, stu1.sex, stu1.height, stu1.
weight);
    /* 实际上,可以用一条赋值语句 stu1 = stu3;即可 */

    /* 初始化 person2 */
    p = &stu1;
    strcpy(p->name, "Wang Wu");
    p->age = 19;
    p->sex = 'M';
    p->height = 170;
    p->weight = 65;
    printf("    %s %d %c %d %d\n", p->name, p->age = 20, p->sex, p->height,
p->weight);
} //End of main
```

3. 函数的参数传递

函数是组成 C 语言程序的基本单位。函数具有相对独立的功能,可以被其他函数调

用,也可调用其他函数。直接或间接的调用自身的函数称为递归函数。C 语言的源程序必须有一个主函数 main(),还可包括若干个自定义函数。函数的设计和调用是程序设计必不可少的技能,是程序设计最重要的基础。能否熟练地设计和使用函数,是体现学生程序设计能力高低的基本条件。这里只对 C 语言函数的基本概念进行简单的总结。

1) 函数的定义

函数定义的一般格式:

```
类型名  函数名(形式参数表)
{  函数体; }
```

其中,类型名是函数返回值的数据类型,如 int、float 等。该值是在函数运行时所得到的某个结果。例如:

```
float  funx(形参数表)
{ 函数体; }
```

表明该函数 funx()在被调用并执行完成时有一个 float 型数值返回。

函数的类型还可以是 void,表明函数不需要返回值。例如:

```
void  funy(形参数表)
{ 函数体; }
```

例 1.1 下面程序中有两个自定义函数和一个主函数。其中,一个自定义函数 sumabc() 计算三个整数之和,另一个自定义函数 displayLine() 仅输出一条线。主函数 main()调用两个函数。

```
#include <stdio.h>
int sumabc(int a, int b, int c)          /* 求三个整数之和 */
{  int  s;
    a = b + c;
    s = a + b + c;
    return s;
}
void displayLine(void)
{ printf("-----\n");
}
void main( )
{  int  x,y, z ,sabc;
    x = y = z = 8;
    display();                          /* 画一条线 */
    printf("\n sum = %d",sumabc(x,y,z)); /* 在输出语句中直接调用函数 sumabc( ) */
    printf("\n   %6d %6d %6d",x,y,z);
    display();                          /* 画一条线 */
    x = 2; y = 4; z = 6;
    sabc = sumabc(x, y, z);              /* 在赋值语句中调用函数 sumabc( ) */
    printf("\n   sum = %d", sabc);
    printf("\n   %6d %6d %6d",x,y,z);
    display();                          /* 画一条线 */
}
```

运行结果:

```
-----
sum = 32
    8    8    8
-----
```

```
-----
sum = 20
    2    4    6
-----
```

2) 函数的参数传递

函数在被调用时,由主调函数提供实参,并传递给形参。在调用结束后,有时可通过参数返回新的数据给主调函数,以实现数据在主调函数和被调函数之间双向传递。C语言实现参数传递的方法分为传值和传地址两大类。

例 1.1 中函数 `sumabc()` 被主函数 `main()` 调用,是通过传值调用,也就是说将 `x`、`y` 和 `z` 的值分别赋值给 `a`、`b` 和 `c`。传值调用只能单向传递,即从主调函数到被调函数。被调函数返回后,在主函数中输出实参的值仍与调用之前相同,不论在调用期间形参值是否改变,调用结束返回到主调函数之后,实参值都不会改变,如 `x` 的值没有改变。

在 C 语言中采用指针变量做形参来实现传址。传址调用的主要特点是可以实现数据双向传递,在调用时实参将地址传给形参,通过地址将对应的数据代入被调函数。如果在调用期间形参值被改变(即该地址对应的单元中的数据发生变化),那么调用结束返回主调函数之后,因为实参地址没有改变,所以其对应的单元中被改变了的数据得以返回,从而实现了数据的双向传递。

例 1.2 将例 1.1 中的函数 `sumabc()` 修改为:

```
int sumabc(int *a, int b, int c)
{
    int s;
    *a = b + c;
    s = *a + b + c;
    return s;
}
```

在 `main()` 中调用 `sumabc()` 函数的语句为:

```
//第一次调用
x = y = z = 8;
printf("\n sum = %d", sumabc(&x, y, z));
printf("\n   %6d%6d%6d", x, y, z);

//第二次调用
x = 2; y = 4; z = 6;
sabc = sumabc(&x, y, z);
printf("\n   sum = %d", sabc);
printf("\n   %6d%6d%6d", x, y, z);
```

则 `x` 的值在调用之前发生了改变。

运行结果:

```
-----  
sum = 32  
    16    8    8  
-----  
sum = 20  
    10    4    6  
-----
```

函数的设计和运用是 C 语言程序设计课程中重要的一部分内容,也有一定的难度,尤其值得重点关注。

预备知识实验

1. 实验目的

- (1) 预习 C 语言中结构体类型的定义方法,掌握数据元素类型的定义方法。
- (2) 掌握存储空间分配和释放的基本方法。
- (3) 巩固结构体变量成员的“.”引用和“->”引用。
- (4) 掌握 C 语言函数的声明、定义和调用。

2. 实验内容

学生信息管理:某年级共有 n 个学生,每个学生的信息构成一个记录,如表 1-1 所示。学生的学号作为关键字(key)。这 n 个记录构成了一个长度为 n 的线性表。现要求完成如下操作:

表 1-1 学生记录

学号	姓名	性别	英语成绩	电路成绩	高数成绩	总分	平均分
4256007	贝小鹏	男	76	68	84		
4256012	黄坚	女	87	64	66		
4256020	林继磊	女	78	69	78		
4256021	潘耀	男	54	66	52		
4256040	潘军均	女	90	89	95		
4256041	钱宇	男	88	77	62		
4256047	陈大武	女	94	95	90		
4256049	陈秀	女	65	82	72		
4256052	陈小名	女	71	84	68		
4256071	施学毅	女	55	63	53		

- (1) 输入所有学生的信息,建立学生表。
- (2) 输出所有学生的信息。
- (3) 查找:在学生表中查找指定学生的信息,如果找到就显示该生的所有信息,否则显示“该学生的信息不存在,查找失败”。
- 例如,给定学号 4256040 和 4256071;再如:给定学号 4256089(表中不存在)。
- (4) 修改:在学生表中修改指定学生的信息,然后显示所有学生信息,如果表中不存在

该生就显示“该学生的信息不存在,修改失败”。

例如,将 4256040 的高等数学改为 88,将 4256099(不存在)的高等数学改为 79。

(5) 添加:在当前学生表的末尾添加一个记录(使记录数增加 1)。

例如,添加的记录为:

4256090 武强 男 82 85 90

要求:按下列方法和步骤完成程序,并上机调试通过。

(1) 用 struct 和 typedef 语句定义学生记录类型(即定义数据元素的类型)。

方法一:设学生的学号、姓名、性别、英语成绩、电路成绩、高数成绩、总分和平均分分别用 number、name、sex、english、circuit、math、total 和 average 表示。这几个数据项(字段)的数据类型依次为 int、字符数组(长度 20)、char、int、int、int、int 和 float。

学生记录的类型定义为 STUDENT。请填写如下空格,以完成定义。

```
typedef struct student {
    int number;           //学号
    char name[20];        //姓名
    char sex[2];          //性别
    _____ ① _____; //英语
    _____ ② _____; //电路
    _____ ③ _____; //高等数学
    int total;            //总分
    float average;        //平均分
}STUDENT;
```

方法二:设学生的学号、姓名、性别分别用 number、name、sex 表示,英语成绩、电路成绩、高数成绩依次存储在一个 int 型数组中,总分和平均分分别用 total 和 average 表示。为了与方法一相区别,现定义学生记录的类型为 STUDENT1:

```
typedef struct student {
    int number;           //学号
    char name[20];        //姓名
    char sex[2];          //性别
    int score[3];         //依次存储英语、电路和数学成绩
    int total;            //总分
    float average;        //平均分
}STUDENT1;
```

(2) 确定存储结构,现采用顺序存储结构。以方法一为例,定义一个类型为 STUDENT 的一维数组 stu,最大元素个数为 100。

```
#define MAXSIZE 100
STUDENT stu[MAXSIZE];
```

(3) 函数声明。

根据题意,可确定如下几个函数:

① 建立学生记录表:

```
void InputStu(STUDENT stu[ ], int n);           //输入,建表,n为输入记录的个数
```

② 输出所有学生记录:

```
void OutputStu(STUDENT stu[ ], int n);           //输出,遍历,n为记录的个数
```

③ 查找:

```
void SearchStu(STUDENT stu[ ], int n);          //查找学生信息
```

④ 修改:

```
void UpdateStu(STUDENT stu[ ], int n);          //修改某记录的某字段
```

⑤ 添加(注意添加后,表中记录数要增加1):

```
void AppendStu(STUDENT stu[ ], int *pn);        //在学生表尾部添加一记录,pn为n的地址
```

(4) 函数调用。

```
int main()
{ //在 main 函数中调用上述声明的函数
    int n;                               //记录个数
    STUDENT stu[MAXSIZE];                //顺序存储结构,方法一 静态一维数组
    /*
        顺序存储结构,方法二 动态一维数组,用 malloc 函数分配如下:
        STUDENT * stu;
        stu = ( STUDENT * ) malloc(sizeof(STUDENT) * MAXSIZE); //内存空间的分配
        注意: 分配空间可用 malloc()函数, 释放空间用 free()函数,如 free(stu);
    */
    int index;
    printf("\n 请输入学生记录个数 n = ");
    scanf("%d", &n);
    InputStu(stu, n);                     //预先处理输入, 建表
    while(1)                              //永真循环,重复显示菜单, 直至退出
    {
        printf("\n ***** 学生信息管理主菜单 ***** \n");
        printf("\t1. 显示学生信息\n");
        printf("\t2. 查找学生信息\n");
        printf("\t3. 修改学生信息\n");
        printf("\t4. 添加学生信息\n");
        printf("\t5. 退出\n\n");
        printf("\t\t 请选择(1~5): ");
        scanf("%d", &index);
        printf("\n ***** \n");
        switch(index) {
            case 1:    OutputStu(stu,n); break;
            case 2:    SearchStu(stu,n); break;
            case 3:    UpdateStu (stu,n); break;
            case 4:    AppendStu (stu,&n); break;
            case 5:    return 0;
            default:    printf("\n 输入有误,请重新输入! \n");
        } //switch
    } //while(1)
} //main
```