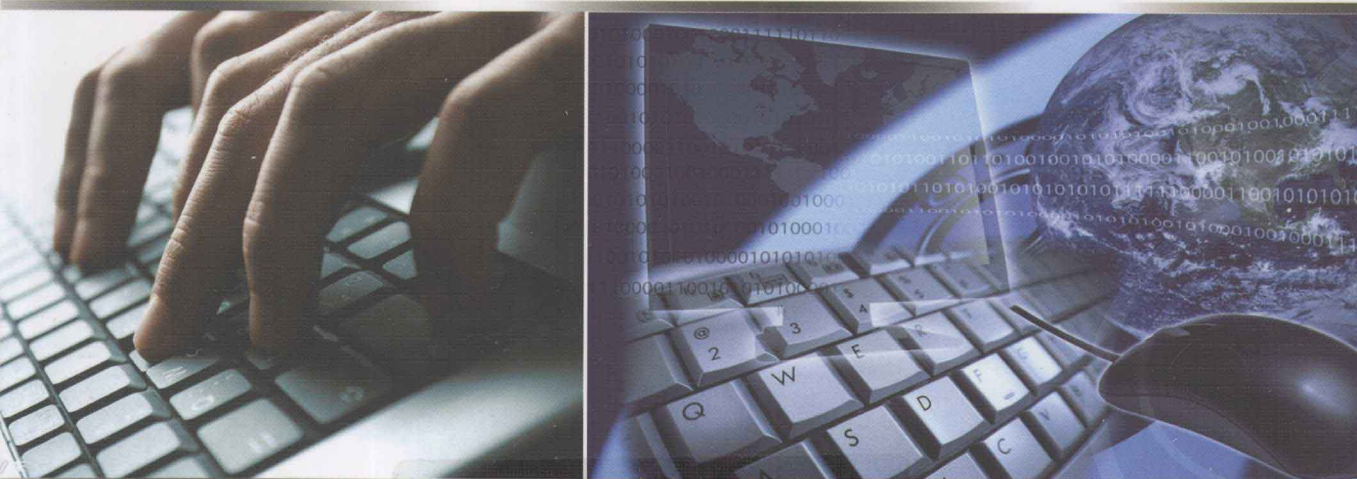




普通高等教育“十二五”规划教材

C语言程序设计实践教程



中国水利水电出版社
www.waterpub.com.cn

普通高等教育“十二五”规划教材

C 语言程序设计实践教程



中国水利水电出版社
www.waterpub.com.cn

内 容 提 要

本书是与《C 语言程序设计》(甘勇主编, 中国水利水电出版社出版) 配套使用的学习用书, 内容对应于主教材中的每一个章节。本书每章分为程序验证、举一反三和程序实例三个部分。程序验证部分给出了每次上机实验的目的和要求、重点和难点、实验内容和课后思考; 举一反三部分针对本章的知识点进行扩展, 提供了针对同一问题不同的解决方法 and 思路; 程序实例部分则通过具体实例对知识点内容加以巩固和提高; 书后附有全国计算机等级考试(二级 C 语言) 简介及 ACM/ICPC 竞赛和在线测试系统介绍。

本书适合作为高校相关课程的实践环节教材, 也适合于各种培训和编程爱好者及参加全国计算机等级(二级 C 语言) 考试人员的自学参考书。

图书在版编目(CIP)数据

C语言程序设计实践教程 / 张宝剑, 肖乐主编. --
北京: 中国水利水电出版社, 2011. 3
普通高等教育“十二五”规划教材
ISBN 978-7-5084-8406-8

I. ①C… II. ①张… ②肖… III. ①
C语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆CIP数据核字(2011)第021130号

策划编辑: 雷顺加/向辉 责任编辑: 杨元泓 加工编辑: 樊昭然 封面设计: 李 佳

书 名	普通高等教育“十二五”规划教材 C 语言程序设计实践教程
作 者	主 编 张宝剑 肖 乐 副主编 炎士涛 高国红 李 晔 付俊辉 杨爱梅
出版发行	中国水利水电出版社 (北京市海淀区玉渊潭南路 1 号 D 座 100038) 网址: www.waterpub.com.cn E-mail: mchannel@263.net (万水) sales@waterpub.com.cn 电话: (010) 68367658 (营销中心)、82562819 (万水)
经 售	全国各地新华书店和相关出版物销售网点
排 版	北京万水电子信息有限公司
印 刷	北京市天竺颖华印刷厂
规 格	184mm×260mm 16 开本 10 25 印张 276 千字
版 次	2011 年 3 月第 1 版 2011 年 3 月第 1 次印刷
印 数	0001—5000 册
定 价	20 00 元

凡购买我社图书, 如有缺页、倒页、脱页的, 本社营销中心负责调换
版权所有·侵权必究

前 言

本书针对现代教育教学改革理念，在提高教学效率的同时，力求提高学生综合实践的能力。本书是在作者多年软件开发和 C 程序设计教学实践经验的基础上，根据现代高校教学改革特有的情况及现代计算机教学的规律，收集分析了大量的教学文献，并基于实际应用而编写的。本书可作为与《C 语言程序设计》（甘勇主编）配套使用的学习用书。

本书每章分为程序验证、举一反三和程序实例三个部分。程序验证部分主要针对简单程序设计、C 语言标准库函数、选择结构、循环结构、函数、数组、字符串与字符数组、结构和联合、指针高级应用、位运算和文件等主要知识点，给出每次上机实验的目的和要求、重点和难点、实验内容和课后思考，上机内容中强调基本知识点的掌握和基本技能的训练，思考题则针对程序验证例题，提出了完善和变通题目要求的思考，以加深对知识点的理解。每小节的程序验证例题，均给出了程序流程图、参考程序和针对该题目的分析，有利于进一步理解和掌握相关知识。举一反三部分针对本章的知识点进行扩展，提供了针对同一问题不同的解决方法和思路。程序实例部分则通过具体实例对知识点内容加以巩固和提高。

本书由张宝剑、肖乐任主编，炎士涛、高国红、李晔、付俊辉、杨爱梅任副主编。参加编写的还有尚展垒、张彦伟、王鹏远、沈高峰、范乃梅等。

感谢郑州轻工业学院、华北水利水电学院、河南工业大学、河南科技学院和中国水利水电出版社对本书出版的大力支持。

由于教学任务繁重，加之本书编写时间紧迫，书中难免会出现一些错误和不足之处，在此恳请广大读者批评指正，并提出宝贵意见。

编 者

2011 年 2 月

目 录

前言

第 1 章 程序设计技术概述	1	第 8 章 字符串与字符数组	95
1.1 算法	1	8.1 内容回顾	95
1.2 Visual C++集成环境	3	8.2 程序验证	97
第 2 章 基本数据类型	7	8.3 程序实例	103
2.1 内容回顾	7	第 9 章 指针进阶	105
2.2 程序验证	8	9.1 内容回顾	105
2.3 举一反三	12	9.2 程序验证	105
2.4 程序实例	14	9.3 程序实例	112
第 3 章 C 语言标准库函数及顺序结构程序设计	16	第 10 章 结构和联合	114
3.1 程序验证	16	10.1 程序验证	114
3.2 举一反三	26	10.2 举一反三	126
3.3 程序实例	28	10.3 程序实例	126
第 4 章 选择结构程序设计	30	第 11 章 位运算	129
4.1 程序验证	30	11.1 程序验证	129
4.2 举一反三	38	11.2 举一反三	131
4.3 程序实例	41	11.3 程序实例	132
第 5 章 循环控制语句	45	第 12 章 文件	133
5.1 程序验证	45	12.1 程序验证	133
5.2 举一反三	66	12.2 举一反三	139
5.3 程序实例	68	12.3 程序实例	141
第 6 章 函数	72	附录一 全国计算机等级考试(二级 C 语言) 简介	143
6.1 内容回顾	72	附录二 ACM/ICPC 竞赛和在线测试系统介绍	150
6.2 程序验证	75	附录三 C 运算符的优先级与结合性	159
第 7 章 数组	84	参考文献	160
7.1 内容回顾	84		
7.2 程序验证	85		

第 1 章 程序设计技术概述

通过对教材第 1 章各小节内容的学习，大家应该对程序设计语言的分类及特点、算法的概念及特性，以及软件的编制步骤有了深入的了解。在本章实验中，将结合 Visual C++ 编程环境，对描述算法的流程图以及利用 Visual C++ 编制程序的具体实现步骤进行更深入的讲解，通过实例达到彻底掌握该部分内容的效果。

1.1 算法

算法（Algorithm）是一系列解决问题的清晰指令，算法代表着用系统的方法描述解决问题的策略机制。也就是说，能够对一定规范的输入，在有限时间内获得所要求的输出。如果一个算法有缺陷，或不适合于某个问题，执行这个算法将不会解决这个问题。

算法的表示方法有多种，包括自然语言、流程图、N-S 图、伪代码和使用计算机语言描述，其中使用图形较为直观，便于理解，所以也是实际应用最多的一种。流程图由一些特定意义的图形、流程线及简要的文字说明构成，它能清晰明确地表示程序的运行过程，流程图中的主要符号如图 1.1 所示。

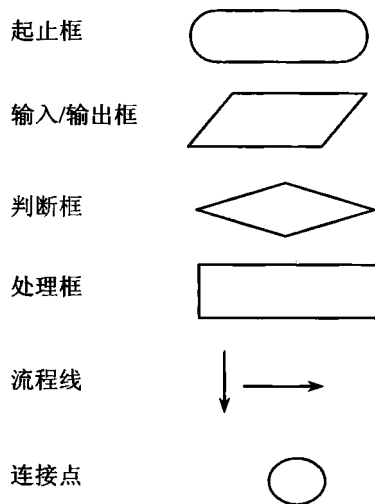


图 1.1 流程图中的主要符号

N-S 图也被称为盒图或 CHAPIN 图，是一种新的流程图，它把整个程序写在一个大框图内，这个大框图由若干小的基本框图构成，这种流程图简称 N-S 图，其基本结构如图 1.2 所示。

1. 实验目的和要求

- (1) 了解表示算法的一般方法。
- (2) 掌握使用流程图描述算法。
- (3) 掌握使用 N-S 图描述算法。

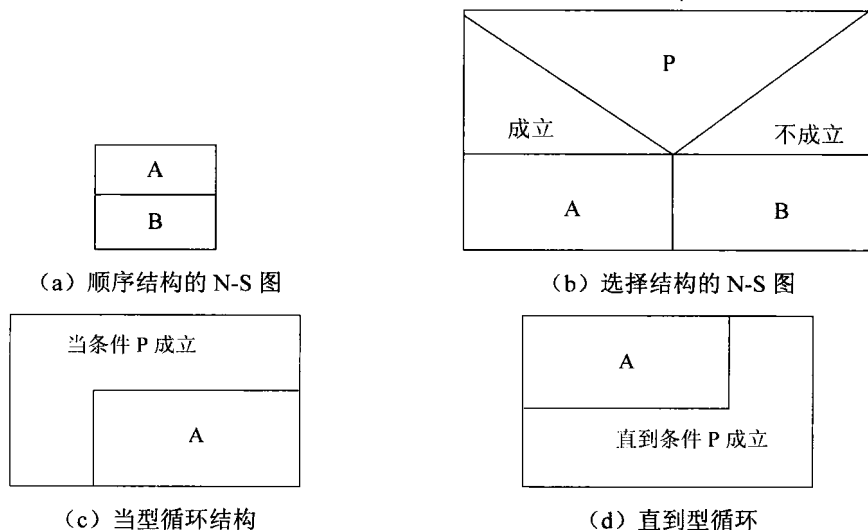


图 1.2 N-S 图基本结构

2. 实验重点和难点

- (1) 各种图形符号的适用条件。
- (2) 分支结构如何实现。

3. 实验内容

输入任意三个整数，求它们的和及平均值。

流程图如图 1.3 所示。

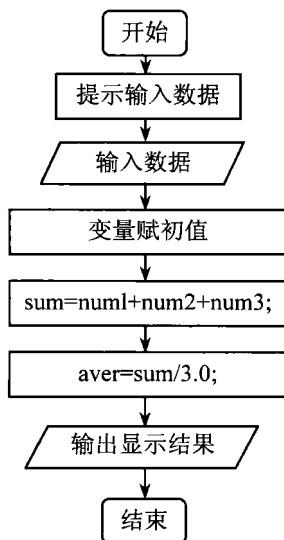


图 1.3 程序运行流程图

参考程序

```
#include <stdio.h>
void main()
{
    int num1,num2,num3,sum;
    float aver;
```

```
printf("Please input three numbers:");  
scanf("%d,%d,%d",&num1,&num2,&num3); /*输入三个整数*/  
sum=num1+num2+num3;                  /*求累计和*/  
aver=sum/3.0;                        /*求平均值*/  
printf("num1=%d,num2=%d,num3=%d\n",num1,num2,num3);  
printf("sum=%d,aver=%7.2f\n",sum,aver);  
}
```

分析:

该程序是一个典型的顺序结构流程。要得到运算结果,必须先有操作数据,因此程序开始运行时首先在屏幕上打印出提示要求用户输入,用户输入的数据分别存放在三个变量 num1、num2 和 num3 中。通过计算公式得到和及平均值存放在变量 sum 和 aver 中,按照用户要求的格式输出结果即可,程序运行结果如图 1.4 所示。

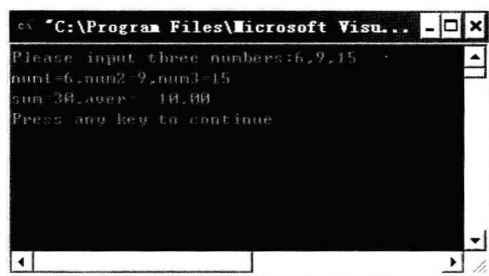


图 1.4 程序运行结果图

课后思考

- (1) 如果要求用 N-S 图画流程,应怎样实现?
- (2) 查阅资料,了解语句#include <stdio.h>的作用?

1.2 Visual C++集成环境

在 Visual C++环境下编程,主要任务是按照题目要求,遵循 C 语言语法规则编写源程序。源程序由字母、数字及其他符号等构成,在计算机内部用相应的 ASCII 码表示,并保存在扩展名为“.C”的文件中。源程序是无法直接被计算机运行的,因为计算机的 CPU 只能执行二进制的机器指令,这就需把 ASCII 码的源程序先翻译成机器指令,然后计算机的 CPU 才能运行翻译好的程序。

源程序翻译过程由两个步骤实现:编译与连接。首先对源程序进行编译处理,即把每一条语句用若干条机器指令来实现,以生成由机器指令组成的目标程序。但目标程序还不能马上交计算机直接运行,因为在源程序中,输入、输出以及常用函数运算并不是用户自己编写的,而是直接调用系统函数库中的库函数。因此,必须把“库函数”的处理过程连接到经编译生成的目标程序中,生成可执行程序,并经机器指令的地址重定位,便可由计算机运行,最终得到结果。

C 语言程序的调试、运行步骤如图 1.5 所示。

Visual C++是微软公司开发的面向 Windows 的 C++语言工具,它不仅支持 C++语言的编程,也兼容 C 语言的编程,因此 Visual C++被广泛地用于各种编程。这里简要地介绍如何在

Visual C++下实现该程序。

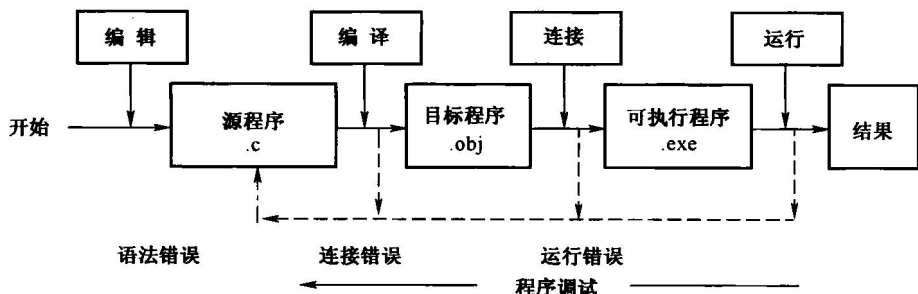


图 1.5 C 语言程序调试、运行步骤图

1. 实验目的和要求

- (1) 了解程序调试运行步骤。
- (2) 掌握 Visual C++编程环境。
- (3) 掌握调试程序错误的步骤和方法。

2. 实验重点和难点

- (1) 在 Visual C++下建立项目、程序文件。
- (2) 对程序运行前出现的错误进行调试。

3. 实验内容

在显示器屏幕上输出欢迎文字“WELCOME”。

解题步骤

(1) 选择“开始”→“程序”→“Microsoft Visual Studio 6.0”→“Microsoft Visual C++ 6.0”命令，启动 Visual C++，屏幕上将显示如图 1.6 所示的窗口。

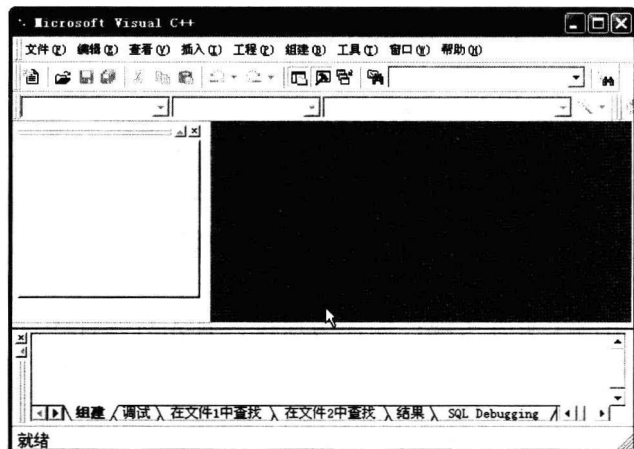


图 1.6 Visual C++启动窗口

(2) 选择“文件”→“新建”命令，弹出如图 1.7 所示的新建工程对话框，选择“Win32 Console Application”选项，并在工程名称中写入工程名“1_1”，然后单击“确定”按钮。

(3) 在弹出的向导对话框中选择“一个空工程”选项，单击“完成”按钮建立该工程，如图 1.8 所示。

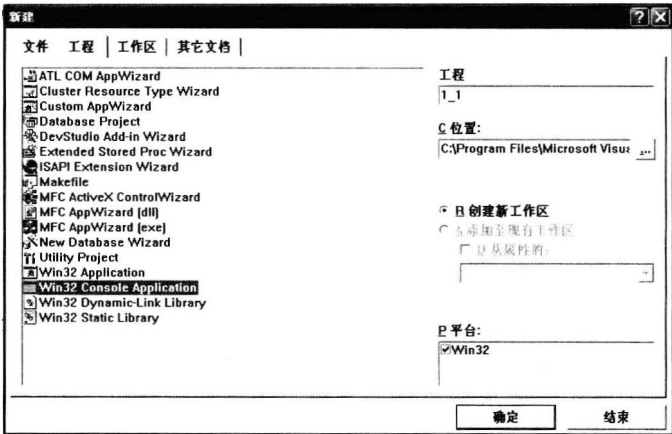


图 1.7 新建工程对话框

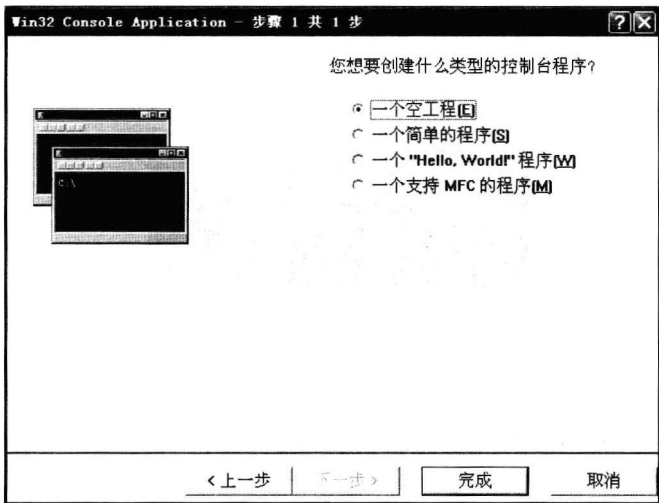


图 1.8 向导对话框

(4) 再次选择“文件”→“新建”命令，弹出如图 1.9 所示的新建文件对话框，选择“C++ Source File”选项，并在“文件名”文本框中输入文件名“1_1”，然后单击“确定”按钮。

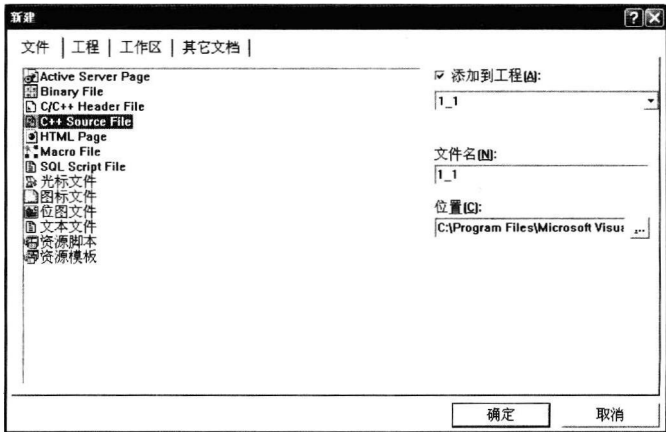


图 1.9 新建文件对话框

(5) 在编辑窗口中输入程序。

参考程序

```
#include <stdio.h>
int main()
{
    printf("WELCOME\n");
    return 0;
}
```

分析:

在 Visual C++ 环境下编写 C 语言程序时,要先建立工程,然后在工程中添加运行程序需要的 C++ 源文件。由于该实例要求输出欢迎字符,用到输出函数,因此在程序开始位置引入“stdio.h”标准头文件。需要注意的是,在任何一个项目中必须有且只有一个 main() 函数,该函数一般返回一个整型数据,因此函数前加上“int”,并在程序结束时加上语句“return 0”表示程序正常结束。输出函数用 printf(), 最后一个“\n”表示换行。程序执行后结果如图 1.10 所示。

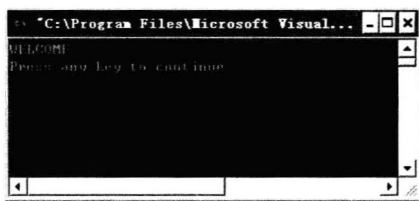


图 1.10 运行结果效果图

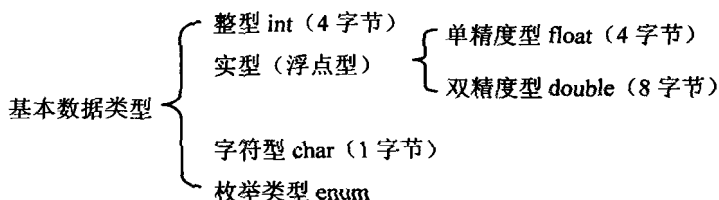
课后思考

- (1) 如果把 main() 函数前的 int 去掉,并且去掉语句“return 0”,程序能否正常运行?
- (2) 如果去掉#include <stdio.h>,程序能否正常运行,为什么?

第 2 章 基本数据类型

2.1 内容回顾

2.1.1 基本数据类型



2.1.2 运算符与表达式

1. 运算符

运算符：表示各种运算的符号，主要分为以下几类：

- 算术运算符：包括+、-、*、/和%。
- 自增、自减运算符：包括++和--。
- 赋值与赋值组合运算符：包括=、+=、-=、*=、/=、%=、<<=、>>=、^=、&=和|=。
- 关系运算符：包括<、<=、>、>=、==和!=。
- 逻辑运算符：包括&&、||和!。
- 位运算符：包括~、|、&、<<、>>和^。
- 条件运算符：包括?和:。
- 逗号运算符：包括,。
- 其他：包括*、&、()、[]、.、->和sizeof。

优先级：指同一个表达式中不同运算符进行计算时的先后次序。

结合性：结合性是针对同一优先级的多个运算符而言的，是指同一个表达式中的多个运算应遵循的运算顺序。分为左结合性和右结合性。

2. 表达式

表达式：使用运算符将常量、变量、函数连接起来的式子。主要分为以下几类：

- 算术表达式：用算术运算符和括号将运算对象（也称操作数）连接起来的、符合 C 语法规则的式子。
- 关系表达式：其结果为逻辑值，逻辑值只有两个值，即逻辑真与逻辑假。
- 逻辑表达式：在求解时，只有必须执行下一个逻辑运算符才能求出表达式的值时，才执行该运算符。
- 赋值表达式：一般形式为**变量 赋值符 表达式**

其求解过程如下：

- (1) 先计算赋值运算符右侧的“表达式”的值。

(2) 将赋值运算符右侧“表达式”的值赋值给左侧的变量。

(3) 整个赋值表达式的值就是被赋值变量的值。

- 逗号表达式：一般形式为表达式 1, 表达式 2, ..., 表达式 n。

逗号表达式的求解过程是自左向右，求解表达式 1，求解表达式 2，…，求解表达式 n。整个逗号表达式的值是表达式 n 的值。

- 条件表达式：一般形式为表达式 ? 表达式 2 : 表达式 3

条件表达式的操作过程：如果表达式 1 成立，则表达式 2 的值就是此条件表达式的值；否则，表达式 3 的值就是此条件表达式的值。

2.1.3 不同数据类型之间的转换

1. 自动类型转换

自动类型转换也称为隐式转换，是计算机按照默认规则自动进行的。

2. 强制类型转换

强制类型转换也称为显式转换，一般形式为：(类型)表达式

2.2 程序验证

1. 实验目的和要求

- (1) 正确使用各种类型的数据。
- (2) 掌握各种数据类型的输入、输出形式。
- (3) 掌握各种运算符和表达式的使用。
- (4) 掌握不同数据类型之间的转换。
- (5) 不同数据类型之间可以转换的条件。

2. 实验重点和难点

- (1) 编写并调试程序。
- (2) 调试程序的注意事项、上机编写 C 语言程序的步骤及错误修改。

3. 实验内容

- (1) 编写程序，已知圆半径 radius=1.25，求圆周长和圆面积。

流程图如图 2.1 所示。

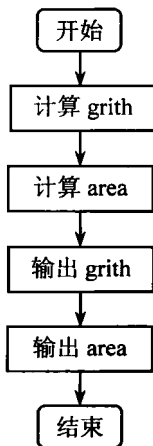


图 2.1 程序流程图

参考程序

```
#include<stdio.h>
main()
{
    float pi=3.14;          /*圆周率*/
    float radius=1.25;       /*半径*/
    float grith=0;           /*周长*/
    float area=0;            /*面积*/
    grith=2*pi*radius;       /*计算周长*/
    area=0.5*pi*radius*radius; /*计算面积*/
    printf("grith= %f\n",grith);
    printf("area= %f\n",area);
}
```

分析:

由圆的周长和面积公式: $C=2\pi R$, $S=0.5\pi R^2$ 可知, 若要计算圆周长和面积, 必须知道圆的半径。题目中已给出其半径为 $radius=1.25$, 注意该数据类型为小数, 应定义为 `float` 或 `double`, 接下来直接将其代入计算公式, 进行计算操作就可以了。实验结果如图 2.2 所示。

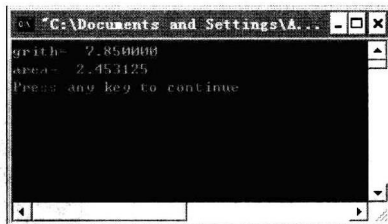


图 2.2 程序运行结果图

(2) 编写程序完成单个字母的大小写转换。

流程图如图 2.3 所示。

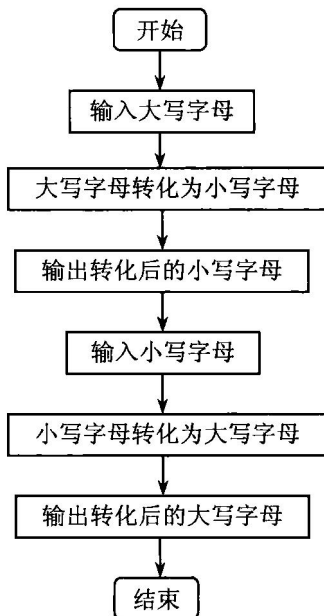


图 2.3 程序流程图

参考实例

```

#include<stdio.h>
main()
{
    char inputLC;        /*输入小写字母*/
    char outputLC;       /*输出小写字母*/
    char inputUC;        /*输入大写字母*/
    char outputUC;       /*输出大写字母*/
    printf("Please input a uppercase char:");
    scanf("%c",&inputUC);
    getchar();
    outputLC=(char)(inputUC+32); /*大写格式转化为小写*/
    printf("After switch to lowercase, the char is:");
    printf("%c\n",outputLC);
    printf("Please input a lowercase char:");
    scanf("%c",&inputLC);
    outputUC=(char)(inputLC-32); /*小写格式转化为大写*/
    printf("After switch to uppercase, the char is:");
    printf("%c\n",outputUC);
}

```

分析:

在 ASCII 码中, 大小写字母所对应的整数值之间的差值为 32, 可以利用这一特点, 进行字母的大小写间的转换, 具体过程是: 将大写字母转化为小写时, 让其加 32; 将小写字母转化为大写时, 减 32 就可以了。实验结果如图 2.4 所示。

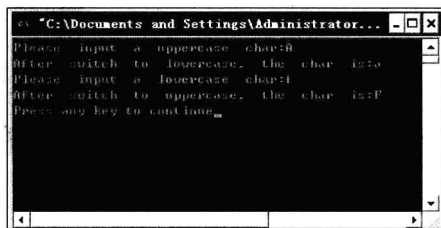


图 2.4 程序运行结果图

(3) 编写程序输出 100 之内的素数。

流程图如图 2.5 所示。

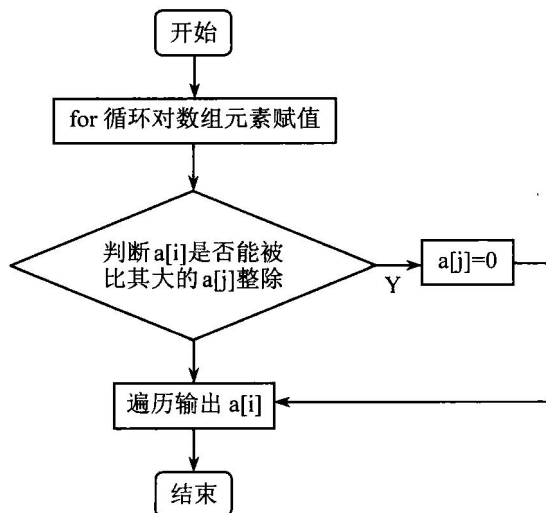


图 2.5 程序流程图

参考实例

```
#include <stdio.h>
#include <math.h>
main()
{
    int i,j,n,a[101];
    for(i=1;i<101;i++)        /*对 a 中各元素赋值*/
    {
        a[i]=i;
    }
    a[1]=0;                    /*去掉 a[1]=1, 因为 1 不是素数*/
    for(i=2;i<sqrt(100);i++)
    {
        for(j=i+1;j<100;j++)
        {
            if(a[i]!=0&& a[j]!=0)
            {
                /*a[j]能被 a[i]整除, 说明其不是素数, 将其值置 0*/
                if(a[j]%a[i]==0)
                {
                    a[j]=0;
                }
            }
        }
    }
    printf("\n");
    for(i=1,n=0;i<=100;i++)    /*输出 a[i]并限制其显示格式*/
    {
        if(a[i]!=0)
        {
            printf("%5d",a[i]);
            n++;
        }
        if(n==10)              /*n=10 时换行*/
        {
            printf("\n");
            n=0;
        }
    }
    printf("\n");
}
```

分析:

首先通过 for 循环, 把 0~100 保存在数组 a 中。把 a 中不是数组的元素设置为 0, 是素数的元素保持不变, 这样在最后遍历 a 并输出其不为 0 的元素就得到了 100 之内的所有元素。对于素数的判断, 可以使用双重循环获取 a 中的不同两个元素 a[i]、a[j], 并确保 a[j]>a[i], 当 a[j]%a[i]=0 时, a[j]就不是素数, 将其值设为 0, 以此循环, 到最后 a 中的非零元素就是所要

的素数。运行结果如图 2.6 所示。

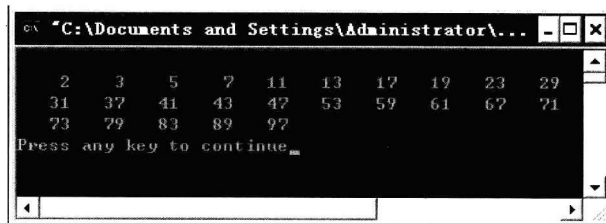


图 2.6 程序运行结果图

课后思考

- (1) 实验内容 (2) 中如果不进行 char 与 int 之间的数据类型转换, 结果如何?
- (2) 实验内容 (2) 中为什么要在代码中添加 getchar()?
- (3) 请考虑实验内容 (3) 中为什么要用 “a[i]<sqr(100)” 限制?

2.3 举一反三

编程实现不同数据类型之间的转换。

```
#include<stdio.h>
main()
{
    int input; /*所输入的整数*/
    char output1;
    float output2;
    double output3;
    printf("Please input a integer between 0 and 127:");
    scanf("%d",&input);
    output1=(char)input; /*转化为 char 类型*/
    output2=(float)input; /*转化为 float 类型*/
    output3=(double)input; /*转化为 double 类型*/
    printf("The equal case in char is: %c",output1);
    printf("\nThe equal case in float is: %f",output2);
    printf("\nThe equal case in double is:%f\n",output3);
}
```

分析:

(1) 该题主要考察数据类型之间的转换, 用强制类型转换就可以了。首先定义一个整型数据 intInput, 用于接受所输入的整数。输入整数 “65”, 如图 2.7 所示。

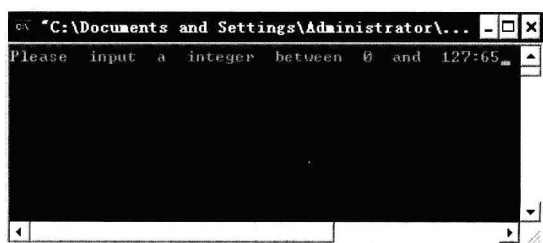


图 2.7 输入数据图