

1

模块 1

初识 C 语言

案例引入——学生成绩管理系统登录界面设计



任务描述

- (1) 安装 C 编译器(如 Visual C++、Dev-C++、Borland Turbo C、Win-TC、Turbo C/C++ for Windows 集成实验与学习环境等),了解所安装 C 编译器的基本使用方法与特点。
- (2) 使用 C 编译器编辑、编译和执行简单 C 程序。
- (3) 设计“学生成绩管理系统”的登录界面程序。



任务目标

- (1) 能够掌握 C 程序的构成。
- (2) 能够应用某种 C 集成编译器进行 C 程序的编辑、编译和执行。
- (3) 能够根据系统提示进行操作、检错和纠错,培养自我学习的能力。
- (4) 能够不断积累程序设计过程中出现的英文,会认、会读、会写和会用。



源代码展示

```
#include <stdio.h>
#include <string.h>
int login() /* 输入密码登录系统,若三次输入不正确,则退出 */
{
    char secretCode[7],ch;
    int i,number=3;
    printf(" \n\n\n");
    printf("\t\t\t* * * * * 欢迎使用学生成绩管理系统 * * * * *\n");
    printf(" \n\n");
```

```
while(number>0)
{
    i=0;
    printf("\n\t\t\t\t 请输入密码:");
    while((ch=getch())!=13)
    {
        secretCode[i++]=ch;
        printf(" * ");
    }
    secretCode[i]='\0';
    if(strcmp(secretCode,"888888")==0)break;
    else
    {
        number--;
        if(number<=0) break;
        printf("\n\n\t\t\t\t 您还有 %d 次机会\n",number);
    }
}
if(number>0) return 1;
else return 0;
}
mainMenu()    /* 系统主菜单,需要以后添加 */
{
    system("cls");
    printf("\n\n\n\t\t\t\t * * * * * 学生成绩管理系统 * * * * * \n");
    getch();
}
main()
{
    if(!login())
    {
        printf("\n\n\t\t\t\t 很遗憾,欢迎您下次再来! \n");
        return 1;
    }
    do
    {
        mainMenu();
    }while(1);
}
```



运行结果

程序运行结果如图 1-1 所示。

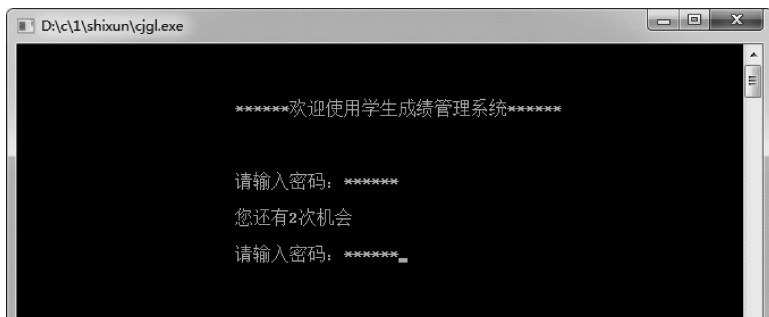


图 1-1 “学生成绩管理系统登录界面设计”程序运行结果



程序分析

本程序由主函数 `main()` 和两个名为 `login()`、`mainMenu()` 的函数构成。`login()` 函数实现密码登录界面设计；`mainMenu()` 是主菜单，需要以后补充。主函数调用两个函数实现相关操作。



知识讲解

C 语言是国际上广泛流行的计算机高级语言，其语法简洁、紧凑，使用方便、灵活，具有丰富的运算符和数据类型，自定义类型也十分灵活。C 程序的模块化主要通过函数实现。由于 C 语言既有高级语言的功能，又有低级语言的一些功能，因此它既可以用来编写系统软件，也可以用来编写应用软件。C 语言在操作系统、工具软件、图形图像处理软件、数值计算、人工智能及数据库系统等多个领域都得到了广泛的应用。目前，许多开发工具，如微软的 Visual C++ 和 C# 及 Java 等都遵循标准 C 语言的基本语法；很多嵌入式系统，多数都采用 C 语言来开发。

1.1 C 语言简介

目前，人们使用的程序设计语言有上百种，它们中的大多数被称为计算机的“高级语言”。C 语言是近年来非常流行的程序设计语言，是一种面向过程的通用程序设计语言，很多人宁愿放弃已经熟悉的其他语言而改用 C 语言。

1.1.1 计算机语言概述

计算机语言的种类非常多，总的来说，可以分成机器语言、汇编语言和高级语言三大类。

1. 机器语言

机器语言是由 0、1 组成的机器指令的集合，是第一代计算机语言。计算机所使用的是

由0和1组成的二进制数,二进制是计算机语言的基础。计算机发明之初,人们只能写出一串串由0和1组成的指令序列交由计算机执行,这种计算机能够认识的语言,就是机器语言。机器语言难读、难记、难写,容易出错,且不同机型互不兼容。

2. 汇编语言

为了减轻使用机器语言编程的烦琐,人们进行了一种有益的改进,用一些简洁的英文字母、符号串来替代一个特定指令的二进制串。例如,用ADD代表加法,用MOV代表数据传递等,使程序比较直观,易于阅读和理解,更容易实现纠错及维护,这种程序设计语言称为汇编语言,即第二代计算机语言。然而计算机是不认识这些符号的,这就需要有一个专门的程序,负责将这些符号翻译成二进制数的机器语言,这种翻译程序被称为汇编程序。

汇编语言同样十分依赖于机器硬件,移植性不好,但效率很高。针对计算机特定硬件而编制的汇编语言程序,能准确发挥计算机硬件的功能和特长,程序精炼且质量高,所以至今仍是一种常用而强有力的软件开发工具。

3. 高级语言

从最初与计算机交流的经历中,人们意识到,应该设计一种语言,这种语言接近于数学语言或人的自然语言,同时又不依赖于计算机硬件,编出的程序能在所有机器上通用。经过努力,1954年,第一个完全脱离机器硬件的高级语言——FORTRAN问世了。这么多年来,共有几百种高级语言出现,有重要意义的有几十种,影响较大、使用较普遍的有FORTRAN、ALGOL、COBOL、BASIC、LISP、PL/1、Pascal、C、PROLOG、C++、VC、VB、Java等。

高级语言的发展也经历了从早期语言到结构化程序设计语言,从面向过程到非过程化程序语言的过程。相应地,软件的开发也由最初的个体手工作坊式的封闭式生产,发展为产业化、流水线式的工业化生产。

高级语言的下一个发展目标是面向应用,也就是说,只需要告诉程序你要干什么,程序就能自动生成算法,自动进行处理,这就是非过程化的程序语言。

4. 高级语言的编译过程

高级语言所编制的程序不能直接被计算机识别,必须经过转换才能被执行,按转换方式可将它们分为两类:解释类和编译类。

(1)解释类:应用程序源代码一边由相应语言的解释器翻译成目标代码,一边执行,因此效率比较低,而且不能生成可独立执行的可执行文件,应用程序不能脱离其解释器,但这种方式比较灵活,可以动态地调整、修改应用程序。

(2)编译类:编译是指在应用源程序执行之前,就将程序源代码翻译成目标代码,因此其目标程序可以脱离其语言环境独立执行,使用比较方便、效率较高。但应用程序一旦需要修改,必须先修改源代码,再重新编译生成新的目标文件才能执行,只有目标文件而没有源代码,修改很不方便。现在大多数的编程语言都是编译型的,如Visual Basic、Visual C++、Visual FoxPro、Delphi等。

1.1.2 C语言概述

1. C语言的发展

在C语言诞生以前,系统软件主要是用汇编语言编写的。汇编语言程序依赖于计算机

硬件,其可读性和可移植性都很差,但一般的高级语言又难以实现对计算机硬件的直接操作。直到1970年,美国贝尔实验室的Ken Thompson设计出了简单而且很接近硬件的B语言,并用B语言写了第一个UNIX操作系统。1972年至1973年,贝尔实验室的D. M. Ritchie在B语言基础上设计出了C语言。1973年,Ken Thompson和D. M. Ritchie合作把UNIX的90%以上用C语言改写,即UNIX第五版。

虽然后来对C语言进行了多次改进,但主要还是在贝尔实验室内部使用。直到1975年UNIX第六版公布后,C语言的突出优点才引起人们的注意。1977年出现了不依赖于机器的C语言编译文本“可移植C语言编译程序”,使C语言移植到其他机器时所需做的工作大大简化,这也推动了UNIX操作系统迅速地在各种机器上的实现。1978年以后,C语言已先后移植到大、中、小、微型机上,已独立于UNIX和PDP。

1983年,美国国家标准化协会(ANSI)根据C语言问世以来各种版本对C语言的发展和扩充,制定了新的标准,称为ANSI C。1987年,ANSI又公布了C语言新标准,即87 ANSI C。1990年,国际标准化组织(ISO)接受了87 ANSI C为ISO C的标准。目前流行的C编译系统都是以它为基础的,本书的叙述基本上以87 ANSI C为基础。

目前流行的C语言的编译器有Microsoft C/C++、Borland C/C++、Visual C++ 6.0、Win-TC、Turbo C/C++ for Windows集成实验与学习环境等,各种版本基本部分是相同的,但略有差异,因此应了解所用计算机系统配置的C编译系统的特点和规定。

2. C语言的特点

C语言之所以能存在和发展,并具有生命力,在于它有不同于其他语言的特点。C语言的主要特点如下。

- (1)简洁、紧凑。C语言一共只有32个关键字,9种控制语句。
- (2)运算符丰富。C语言共有44种运算符。它把括号、赋值、强制类型转换等都作为运算符处理,从而使C语言的运算类型极其丰富,表达式多样化。
- (3)具有丰富的数据类型。C语言具有整型、实型、字符型、数组类型、指针类型、结构体类型等数据类型,能方便地构造更加复杂的数据结构(如链表、树、栈等)。
- (4)C语言是一种结构化的程序设计语言。C语言具有结构化的控制语句(如if、switch、for、while、do... while),用函数作为程序的模块单位,便于实现程序的模块化。
- (5)语法限制不严格,程序设计灵活。例如,C语言不检查数组下标越界,C语言不限制数据转化,不限制指针的使用,程序正确性由程序员保证。灵活和安全是一对矛盾,对语法限制的不严格可能也是C语言的一个缺点,黑客可能使用越界的数组攻击用户的计算机系统。
- (6)能进行位操作,可以直接对部分硬件进行操作。例如,C语言可以直接操作计算机硬件,如寄存器、各种外设I/O端口等;C语言的指针可以直接访问内存物理地址;C语言类似汇编语言的位操作可以方便地检查系统硬件的状态。
- (7)可移植性好。用C语言编写的程序基本上不需要修改或只需要少量修改就可以移植到其他计算机系统或操作系统中。
- (8)C语言编译后生成的目标代码质量高,程序的执行效率高。

1.2 C 程序介绍

C 程序是由 C 语言的若干语句序列组成的。为了了解 C 程序的结构特点,我们先看几个 C 程序,虽然有关内容还未介绍,但可以从这些例子中了解一个 C 程序的基本构成。

1.2.1 C 程序的总体结构

【例 1-1】 输入矩形的两条边长,求矩形的面积。

程序代码如下。

```
#include "stdio.h"           /* 头文件(含输入/输出函数) */
main()                       /* 主函数 */
{
    float a,b,area;          /* 变量声明 */
    scanf("%f %f",&a,&b);    /* 键盘输入数据给变量 */
    area=a*b;                /* 计算 */
    printf("area=%f\n",area); /* 输出变量的值至显示器 */
}
```

【例 1-2】 通过函数调用求两个数中的最大值。

程序代码如下。

```
#include "stdio.h"           /* 头文件(含输入/输出函数) */
int max(int x,int y)         /* 求两整数中较大的数 */
{
    return (x>y? x:y);       /* 返回 x,y 中的最大值,通过 max 带回调用处 */
}
main()                       /* 主函数 */
{
    int a,b,c;               /* 声明部分,定义变量 */
    scanf("%d %d",&a,&b);    /* 键盘输入数据给变量 */
    c=max(a,b);              /* 调用 max,将调用结果赋给 c */
    printf("max=%d",c);     /* 输出变量的值至显示器 */
}
```

注意:【例 1-2】中包括两个函数,即主函数 main 和调用函数 max,max 的作用是求任意两个整数中的最大值。

通过以上例子可以总结出 C 程序结构的主要特点如下。

(1) C 程序是由函数构成的。一个 C 源程序至少包含一个 main 函数,也可以包含一个 main 函数和若干其他函数。所以说,C 是函数式的语言,函数是 C 程序的基本单位。

(2) main 函数(主函数)是每个程序执行的起始点。一个 C 程序总是从 main 函数开始执行,并在 main 函数中结束。main 函数的书写位置是任意的,可以将 main 函数放在整个程序的最前面,也可以放在整个程序的最后,或者放在其他函数之间。



测试
等级考试真题

(3) 一个函数由函数说明和函数体两部分组成。函数结构如下。

函数类型 函数名(形参表)

{

[声明部分]:在这部分定义本函数所使用的变量。

[执行部分]:由若干条语句组成命令序列。

}

当然,在某些情况下也可以没有声明部分,甚至可以既没有声明部分也没有执行部分。

注意: 变量声明部分必须书写在执行部分之前。

(4) C 程序的每个语句,都以分号(;)作为语句结束符。

(5) C 程序书写格式自由,一行可以写几个语句,一个语句可以写在多行上。

(6) 可以用 /* */ 对程序任何部分做注释,以增加可读性。

注释内容要写在“/*”和“*/”之间。注释部分允许出现在程序中的任何位置。注释部分只是用于阅读,对程序的运行不起作用。C 语言中注释不允许嵌套。注释可以用西文,也可以用中文。使用注释是编程人员的良好习惯,注释也是重要的交流工具。

(7) C 语言本身不提供输入/输出语句,输入/输出的操作是通过调用库函数 scanf 和 printf 等来完成的。

1.2.2 C 程序的书写规则

C 语言语句简练、语义丰富、格式灵活,为了提高程序的可读性,应该遵循 C 程序的书写规则,养成良好的书写习惯。

C 程序书写格式的一般要求如下。

(1) 一般一条语句占一行,以分号结束;有些短语句也可以一行多条;长语句也可以写成多行,不加续行符。

(2) 用花括号({})标明程序的层次结构。

(3) 采用逐层缩进格式,使程序清晰易读。通常向右缩进两个字符或一个水平制表符。

(4) 标识符、关键字之间加空格间隔,若已有间隔符则可以不加。

(5) 使用注释信息,增加程序的可读性。

1.3 C 程序的开发过程

C 语言是一种编译型的程序设计语言。用 C 语言开发程序,需要一个开发环境。目前流行的集成环境有 Borland Turbo C 或称 Turbo C、Visual C++、Dev-C++、Win-TC、Borland C++。本节以 Visual C++ 为开发环境介绍 C 程序的上机操作过程。

1.3.1 C 程序的实现过程

从编写一个 C 程序到完成运行得到结果一般需要经过以下几个步骤。

1. 编辑

编辑是将源程序通过键盘逐个字符输入计算机内存,并加以修改,最后以文本文件的形

式保存到磁盘文件中,其文件扩展名为“.c”。

2. 编译

编译是将已编辑好的源程序翻译成二进制的目标代码。在编译时,要对源程序进行语法检查,如发现错误,则显示出错信息,此时应重新进入编辑状态,对源程序进行修改后再重新编译,直到通过编译为止,此时生成扩展名为“.obj”的同名目标文件。

3. 连接

连接是将各个模块的二进制目标代码与系统标准模块经过连接处理后,得到可执行文件,其扩展名为“.exe”。

4. 运行

直接运行可执行文件即可得到程序运行结果。通常,在 DOS 环境下直接输入可执行文件名,在 Windows 环境下双击可执行文件名即可运行程序。

1.3.2 在 Visual C++ 环境下实现 C 程序

1993 年,Microsoft 公司推出 Visual Studio 1.0,此后新版本不断问世。虽然 Visual Studio 经历了诸多版本的改进升级,但是 Visual Studio 6.0 以后的 C++ 没有多大的变化。随着 C++ 新标准的公布,Visual Studio 2010 在 C++ 开发方面带来了许多革命性的变化。由于最新的计算机等级考试(二级 C)的环境改成了 Visual C++ 2010 学习版,下面介绍在 Visual C++ 2010 学习版环境下如何实现 C 程序。

1. Visual C++ 2010 学习版主界面

在 Windows 系统任务栏中,执行“开始”→Microsoft Visual C++ 2010 Express 命令,即可启动 Visual C++ 2010 学习版开发环境,其主界面如图 1-2 所示。

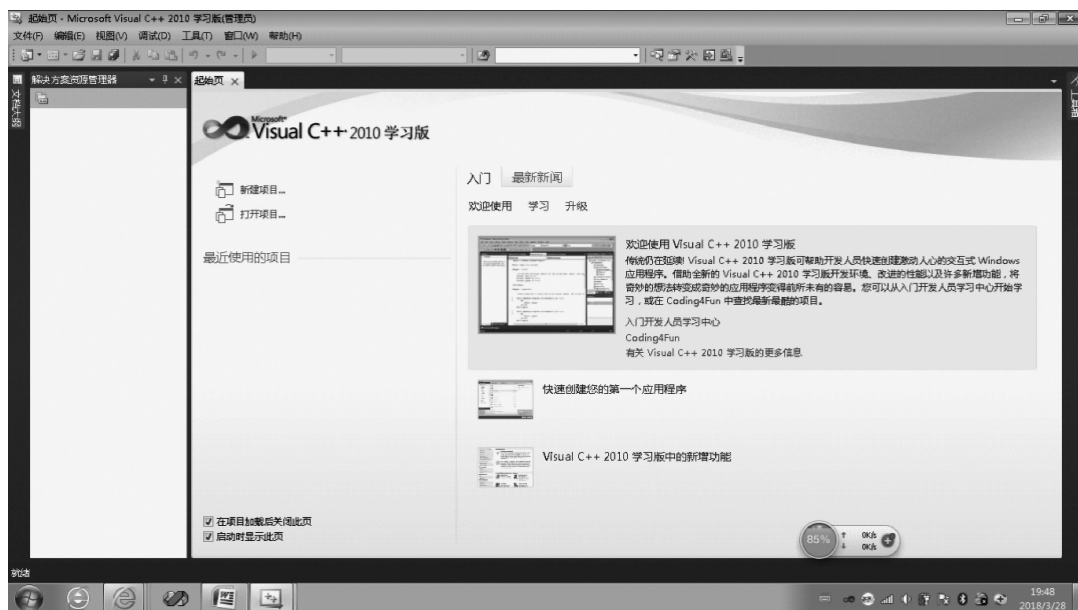


图 1-2 Visual C++ 2010 学习版主界面

2. 在 Visual C++ 2010 学习版中实现 C 程序

1) 创建项目

执行“文件”→“新建”→“项目”命令，弹出“新建项目”对话框，如图 1-3 所示。在“新建项目”对话框中，选择“Visual C++”→“空项目”选项，在下边的“名称”文本框中输入项目名称，如“cp”，选择项目路径，单击“确定”按钮，进入新建项目界面，如图 1-4 所示。



图 1-3 “新建项目”对话框

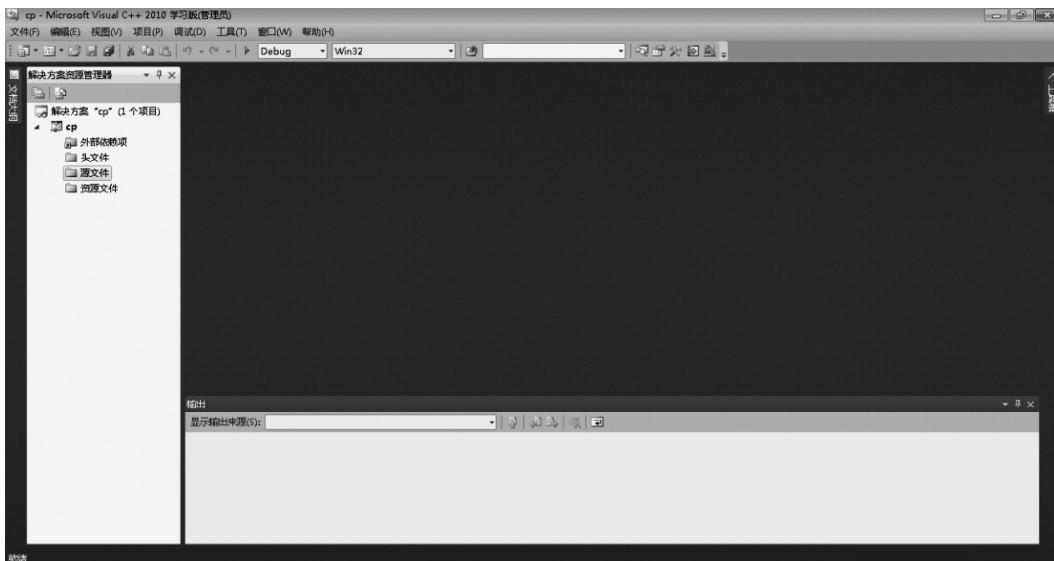


图 1-4 新建项目窗口

2) 建立 C 文件

在新建项目窗口中的右侧找到新建的项目，如“cp”。右击 cp 下的“源文件”，在弹出的快捷菜单中选择“添加”→“新建项”命令，弹出“添加新项”对话框，如图 1-5 所示。

在中间窗格中选择“C++ 文件”选项，在下边“名称”文本框中输入文件名，如“lianxi1.c”，选择文件的路径。单击“添加”按钮，进入代码编辑窗口。在代码编辑窗口中输入“lianxi1.c”的源代码，如图 1-6 所示。



图 1-5 “添加新项”对话框

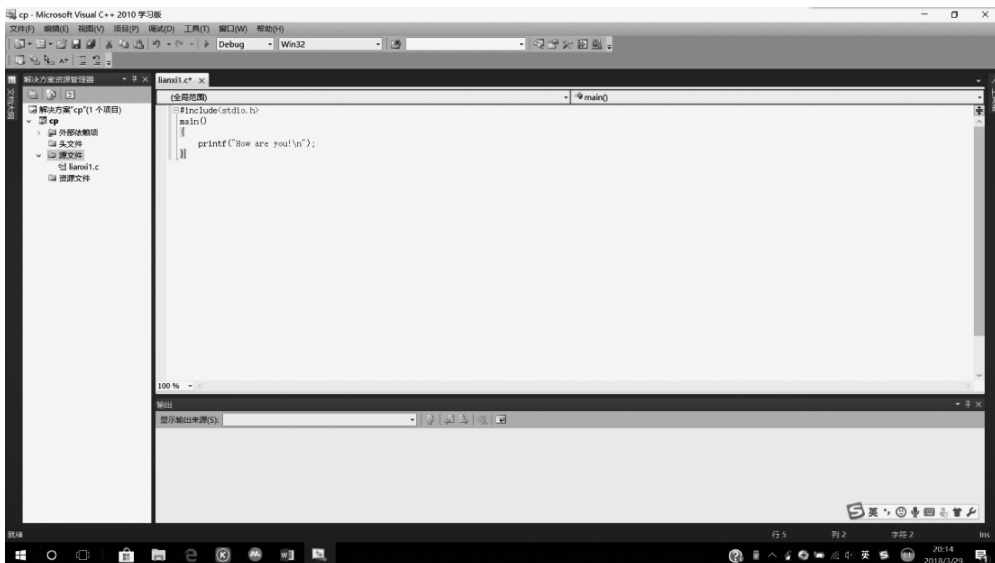


图 1-6 代码编辑窗口

3) 生成并运行程序

将 C 源程序输入结束后,按 Ctrl+F5 组合键,弹出提示对话框,如图 1-7 所示。单击“是”按钮,若程序没有错误则显示如图 1-8 所示的结果。若程序有错误,则在代码编辑窗口下边的输出窗口有提示错误,在编辑窗口根据错误提示修改源代码,按 Ctrl+F5 组合键重新生成运行。



图 1-7 提示对话框

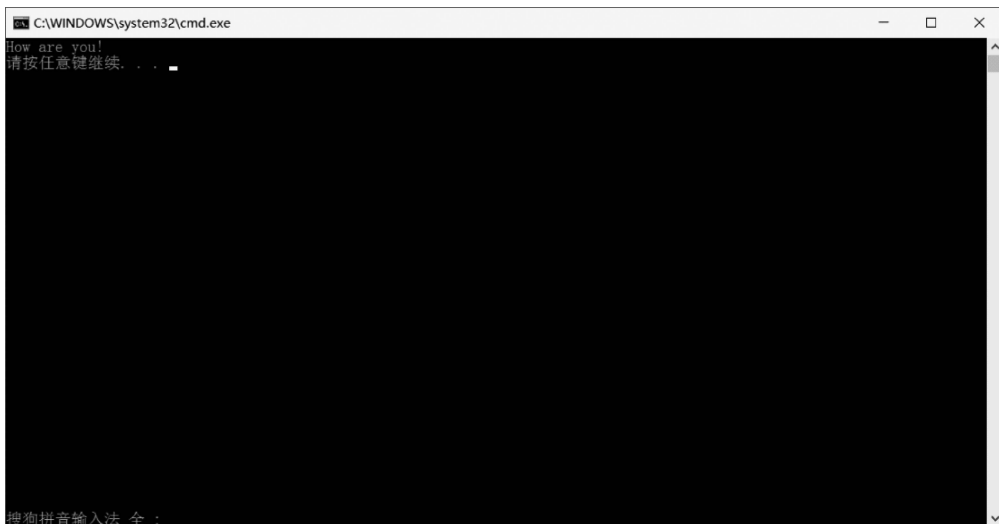


图 1-8 程序运行结果

注意: 按 Ctrl+F5 组合键执行生成运行后,程序结果可能一闪而过无法看到。可以通过下面两种方法显示程序结果。

第一种方法:设置控制台显示。

在代码编辑窗口(见图 1-6)右击“cp”,在弹出的快捷菜单中选择“属性”命令,弹出“cp 属性页”对话框,如图 1-9 所示。在左边一栏找到“配置属性”→“链接器”→“系统”,单击“系统”后,将右边栏的“子系统”值设置为“控制台(/SUBSYSTEM:CONSOLE)”。经过这样的配置后,再按 Ctrl+F5 组合键,程序执行结束就会停留在控制台界面,显示结果并提示“请

按任意键继续...”。

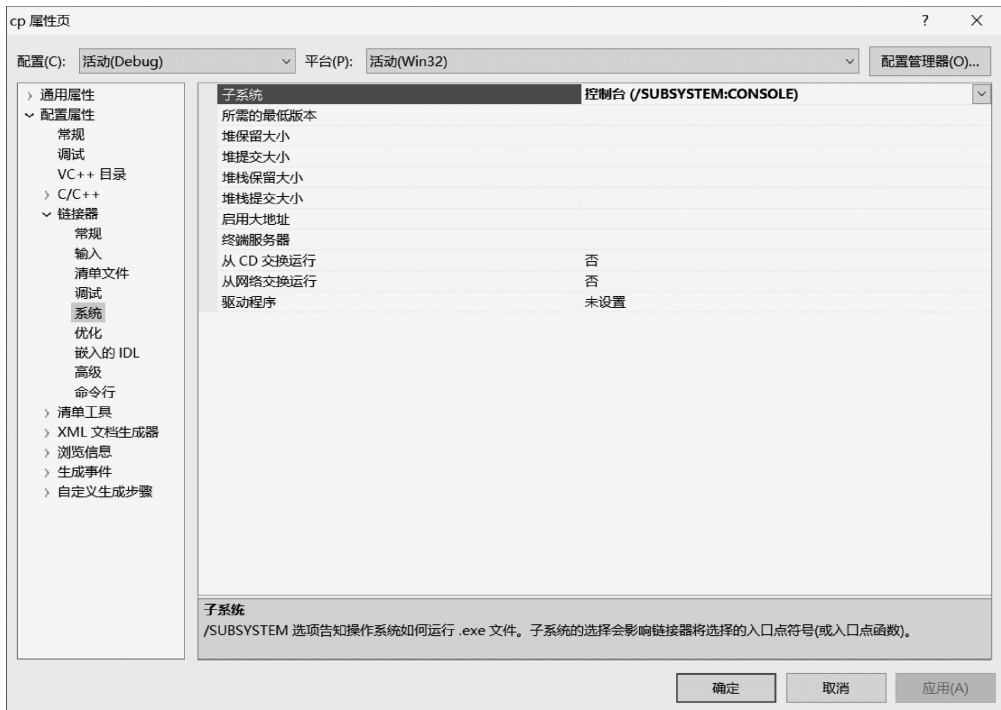


图 1-9 “cp 属性页”对话框

第二种方法:添加代码。可以通过在程序最后添加代码“getch();”或“system(“pause”);”,便可显示程序结果。使用“system(“pause”);”需要包含头文件“stdlib. h”,即“#include<stdlib. h>”。

等级考试重难点讲解与真题解析

本模块主要考查 C 程序的构成与格式、C 程序的开发过程等。从历次等级考试的试题来看,本模块属于非重点考查对象,一般以考查概念性的知识为主。

一、重点、难点解析

1. C 程序的构成

1) C 程序主要由函数构成

每一个 C 程序必须有且仅有一个 main 函数(主函数)。C 程序的执行总是从主函数开始,并在主函数中结束。主函数的书写位置是任意的,它可以调用任何非主函数;任何非主函数都可以相互调用,但是不能调用主函数。

2) 一个函数由两部分构成

(1) 函数的说明部分,包括函数类型(返回值类型)、函数名和形参表。

(2) 函数体。函数体中包含变量声明部分和执行部分。

注意: 变量声明部分必须书写在执行部分之前。

2. C 程序的书写规则

C 语言的书写格式自由,一行内可以写几个语句,一个语句也可以分写在多行。每个语句和数据定义的最后必须有一个分号。C 语言中的注释可以用“/*”和“*/”括起来,注释可以在任何允许插入空格符的地方插入。C 语言中注释不允许嵌套。注释可以用西文,也可以用中文。



视频
注释

3. C 程序的编译与执行

由 C 语句构成的指令序列称为 C 源程序,C 源程序经过 C 语言编译程序编译之后生成一个后缀为“.obj”的二进制文件(称为目标文件);然后由“连接程序”把“.obj”文件与 C 语言提供的各种库函数连接起来生成一个后缀为“.exe”的可执行文件。在 DOS 状态下,输入此文件名(不必输入后缀“.exe”)并按 Enter 键,该文件就可以执行。

二、真题解析

1. 用 FORTRAN 语言编制的源程序要变为目标程序,必须经过()。[2017 年 9 月]
A. 汇编 B. 解释 C. 编辑 D. 编译

解析 高级语言所编制的程序不能直接被计算机识别,必须经过解释或编译转换才能被执行。FORTRAN 语言是一种编译型的程序设计语言。所以 D 是正确的。

2. C 语言程序的基本单位是()。[2017 年 9 月]
A. 程序行 B. 语句 C. 函数 D. 字符

解析 C 程序是由函数构成的。一个 C 源程序可以包含一个 main 函数和若干其他函数,函数是 C 程序的基本单位。所以 C 是正确的。

3. 在计算机内部,一切信息存取、处理和传送的形式是()。[2016 年 9 月]
A. ASCII 码 B. BCD 码 C. 二进制 D. 十六进制

解析 计算机所使用的是由 0 和 1 组成的二进制数,一切信息存取、处理和传送的形式都是二进制,所以答案为 C。

4. 以下叙述中错误的是()。[2011 年 3 月]
A. C 语言的可执行程序是由一系列机器指令构成的
B. 用 C 语言编写的源程序不能直接在计算机上运行
C. 通过编译得到的二进制目标程序需要连接才可以运行
D. 在没有安装 C 语言集成开发环境的机器上不能运行 C 源程序生成的“.exe”文件

解析 开发一个 C 程序要经过编辑、编译、连接和运行 4 个步骤。经过编译过的目标程序和用到的库函数连接生成的可执行程序,在 DOS 环境下直接键入可执行文件名,在 Windows 环境下双击可执行文件名即可运行程序,所以 D 是错误的。

5. 计算机高级语言程序的运行方法有编译执行和解释执行两种,以下叙述中正确的是()。[2011 年 3 月]

A. C 语言程序仅可以编译执行
B. C 语言程序仅可以解释执行
C. C 语言程序既可以编译执行又可以解释执行

D. 以上说法都不对

解析 C语言是一种编译型的程序设计语言。所以A是正确的。



思考与练习

一、选择题

- 下列叙述错误的是()。
 - 计算机不能直接执行用C语言编写的源程序
 - C程序经C编译程序编译后,生成后缀为“.obj”的文件是一个二进制文件
 - 后缀为“.obj”的文件,经连接程序生成后缀为“.exe”的文件是一个二进制文件
 - 后缀为“.obj”和“.exe”的二进制文件都可以直接运行
- 下列叙述中正确的是()。
 - 每个C程序文件中都必须要有个main()函数
 - 在C程序中main()函数的位置是固定的
 - C程序中所有函数之间都可以相互调用,与函数所在位置无关
 - 在C程序的函数中不能定义另一个函数
- C语言源程序名的后缀是()。

A. .exe B. .c C. .obj D. .cpp
- 下列叙述中正确的是()。
 - C程序中的注释只能出现在程序的开始位置和语句的后面
 - C程序书写格式严格,要求一行内只能写一个语句
 - 简单C语句必须以分号结束
 - 用C语言编写的程序只能放在一个程序文件中
- 下列说法正确的是()。
 - 在书写C语言源程序时,每个语句以逗号结束
 - 注释时,“/”和“*”号间可以有空格
 - 无论注释内容多少,在对程序编译时都被忽略
 - C程序的基本组成单位是语句
- 下列叙述正确的是()。
 - 在C程序中无论整数还是实数,只要在允许的范围内都能准确无误地表示
 - 在执行C程序时,不一定从main()函数开始
 - C程序由函数组成
 - C程序由函数和过程组成

二、模仿编程

- 模仿【例 1-1】编写程序,已知一个圆的半径 $r=2.5$,求圆的面积。
- 模仿本模块案例,设计一个银行登录界面程序。

2

模块 2

C 程序数据描述与计算

案例引入——学生成绩管理中数据的描述与计算



任务描述

分析 C 语言对客观事物的数据描述和计算处理。

建立课程小组成员的成绩表。假设学生成绩表如表 2-1 所示,试对各列数据进行适当类型的常量、变量描述与表达式的计算处理。

表 2-1 学生成绩表

no	name	t1	t2	t3	t4	avg	sick	l_e	absent	f_e	score	grade
1	刘延运	80	60	80	60					96.0		
2	孙燕	95	95	90	80		1			86.0		
3	王珊珊	95	98	90	80					95.0		
4	刘斌	90	95	95	99					75.5		
5	王一山	96	95	99	99					64.5		
6	胡志民	98	90	90	85					91.5		
7	张瑜	98	95	99	90					89.5		
8	李冰	95	95	90	80		1			55.0		
9	张珊	85	99	99	99					89.5		
10	陈明明	95	90	95	90					90.0		

- (1)明确采用哪种类型的常量描述数据。
- (2)明确采用哪种类型的变量描述数据。
- (3)明确采用什么样的表达式来处理数据。
- (4)完成 2 位同学 4 个任务(t1,t2,t3,t4)的输入,计算每位同学任务的平均分(avg)并

输出。

任务目标

- (1)能够熟练地根据数据处理需求描述合适数据类型的常量,定义合适数据类型的变量。
- (2)能够熟练地根据数据处理需求正确编写表达式。
- (3)能够应用某种C集成编译器进行C程序的编辑、编译和执行。
- (4)能够根据系统提示进行操作、检错和纠错,培养自我学习的能力。
- (5)培养严肃、严格和严谨的工作作风,表达式编写和代码编写一丝不苟。

源代码展示

```
#include <stdio.h>
main()
{
    int t1,t2,t3,t4,i=1;
    float avg;
    printf("请输入第%d位同学4个任务的成绩:",i);
    scanf("%d%d%d%d",&t1,&t2,&t3,&t4); /* 输入4个任务成绩 */
    avg=(float)(t1+t2+t3+t4)/4;      /* 求4个任务的平均成绩 */
    printf("%d 刘延运 %4d %4d %4d %4d %6.1f\n",i,t1,t2,t3,t4,avg); /* 输出学
生的信息 */
    i++;
    printf("请输入第%d位同学4个任务的成绩:",i);
    scanf("%d%d%d%d",&t1,&t2,&t3,&t4);
    avg=(float)(t1+t2+t3+t4)/4;
    printf("%d 孙燕 %4d %4d %4d %4d %6.1f\n",i,t1,t2,t3,t4,avg);
}
```

运行结果

程序运行结果如图 2-1 所示。

```
D:\c\2\shixun\c\jgl2.exe
请输入第1位同学4个任务的成绩: 80 60 80 60
1 刘延运  80  60  80  60  70.0
请输入第2位同学4个任务的成绩: 95 95 90 80
2 孙燕   95  95  90  80  90.0

-----
Process exited after 27.06 seconds with return value 29
请按任意键继续. . .
```

图 2-1 “学生成绩管理中数据的描述与计算”程序运行结果



程序分析

首先分析成绩表中的各个数据项,学生名字选用字符串原样输出,4个任务成绩(t_1 、 t_2 、 t_3 、 t_4)定义为整型,平均成绩(avg)定义为实型。在计算平均成绩时由于4个任务总分($t_1+t_2+t_3+t_4$)为整型,为了保证平均分得到合理的实数,采用强制转换($float$)($t_1+t_2+t_3+t_4$)。对每个学生的数据处理语句基本一致,用 $i++$ 来计数学生人数。



知识讲解

C语言的数据结构是以数据类型的形式出现的。在C语言中,每个数据都属于一个确定的数据类型,不同类型的数据在数据表示形式、合法的取值范围、占用内存空间大小以及可以参与的运算等方面有所不同。

运算符是程序中完成各种操作的操作码,C语言中运算符的种类很多,本模块重点介绍常用的算术运算符、赋值运算符、自增自减运算符等。

2.1 常量与变量

在C语言中基本数据有常量和变量之分,它们分别属于不同的数据类型。在程序中对用到的所有数据都必须指定其数据类型,常量可以不经说明而直接引用,变量则必须先定义后使用,而要标识一个常量或变量,必须采用合法的标识符。

2.1.1 标识符

任何一种语言,都有自己的符号、单词以及语句的构成规则。C语言作为计算机的一种程序设计语言,也有自己的字符集、标识符以及命名规则。用来标识变量名、符号常量名、函数名、数组名、类型名、文件名的有效字符序列称为标识符。

1. 关键字

C语言系统设置特定含义、专门用途的字符序列称为关键字。关键字不能作为其他用途,只能小写。例如,用来说明变量类型的标识符 int 、 $double$ 以及 if 语句中的 if 、 $else$ 等都有专门的用途,它们不能再用作变量名和函数名。

2. 预定义标识符

所谓的预定义标识符,是指在C语言中预先定义并具有特定含义的标识符,如C语言提供的库函数的名字(如 $printf$)和预编译处理命令(如 $define$)等。C语言允许把这类标识符重新定义另作他用,但这将使这些标识符失去预先定义的原意。鉴于目前各种计算机系统的C语言都一致地把这类标识符作为固定的库函数或预编译处理中的专门命令使用,因此为了避免误解,建议用户不要把这类预定义标识符另作他用。

3. 用户标识符

由用户根据需要定义的标识符称为用户标识符,又称为自定义标识符。用户标识符一

般用来给常量、变量、函数、数组、类型、文件等命名。



测试
等级考试真题

用户标识符命名规则如下。

(1)只能由字母、数字和下划线组成,且第一个字符必须为字母或下划线。

(2)有大小写之分,如 sum、SUM 和 Sum 是 3 个不同的标识符。在 C 程序中,变量名使用小写,常量名用大写,但不绝对。

(3)ANSI C 没有限制标识符长度,但各个编译系统都有自己的规定和限制。有的系统取 8 个字符,Turbo C 则允许 32 个字符。

(4)标识符不能与“关键字”同名,最好也不与“预定义标识符”同名。

(5)标识符命名应当有一定的意义,做到见名知义,以增加程序的可读性。最好使用英文单词及其组合,便于记忆和阅读,尽量少用汉语拼音来命名。例如:

合法的用户标识符:al、x2、s_1、s_2、_3、ggde2f_1。

不合法的用户标识符:df 1、1a、d@sina、s * b、+d。

2.1.2 常量

在程序的运行过程中,其值不能改变的量称为常量。

1. 常量的类型



测试
等级考试真题

在 C 语言中,有整型常量、实型常量、字符型常量和字符串常量等。整型常量还可以进一步分为短整型常量、长整型常量等。

整型常量和实型常量又称为数值型常量,它们有正负之分。基本整型常量只用数字表示,不带小数点,如 12、0、-3 为整型常量。实型常量必须用带小数点的数表示,如 4.6、-1.23 为实型常量。'a'、'A' 为字符型常量,"abc"和"hello"是字符串常量。

2. 符号常量

在 C 程序中可以用一个符号名代表一个常量,称为符号常量。这个符号常量必须在程序中指定,并符合标识符的命名规则。用 define 定义时,前面必须以“#”开头,命令行最后不加分号。

为了区别于一般的变量名,符号常量通常用大写字母。

【例 2-1】 计算圆的面积。

程序代码如下。

```
#include <stdio.h>
#define PI 3.14159
main()
{
    float r,area;
    r=5.0;
    area= PI * r * r;
    printf("area= %f\n",area);
}
```

程序执行结果如下。

```
area=78.539750
```

程序中使用“#define”命令行定义PI代表一串字符3.14159。在对程序编译时,凡程序中出现PI的地方,编译程序均用3.14159来替换。

使用符号常量有如下好处。

- (1)含义清楚,见名知意。
- (2)修改方便,一改全改。

2.1.3 变量

在程序的运行过程中,其值可以改变的量称为变量。程序中用到的所有变量都必须有一个名字作为标识,变量的名字由用户定义。

关于变量的使用,要注意以下几点。

(1)变量名必须符合标识符命名规则,一个变量实质代表内存中的某个存储单元。变量名在程序运行过程中不会改变,变量的值可以改变。

(2)C语言中的变量必须“先定义,后使用”。对变量的定义通常放在函数体内的前部,但也可以放在函数的外部或复合语句的开头。

- ①只有声明过的变量才可以在程序中使用,这使得变量名的拼写错误容易发现。
- ②声明的变量属于确定的类型,编译系统可方便地检查变量所进行运算的合法性。
- ③在编译时根据变量类型可以为变量分配相应的存储空间。
- (3)像常量一样,变量也有整型变量、实型变量、字符型变量等不同类型。

2.2 基本数据类型

C语言提供了丰富的数据类型,其中最常用的是基本类型。下面依次介绍整型数据、实型数据和字符型数据类型。

2.2.1 整型数据

1. 整型数据的分类

C语言中整型数据有短整型(short int)、基本整型(int)、长整型(long int)、无符号型(unsigned)。若不指定变量为无符号型,则变量隐含为有符号型(signed)。

表2-2列出了ANSI标准定义的整型数据类型。Turbo C的规定与此表一致。

表 2-2 整型数据类型

类 型	类 型 名	字 节	取值范围
整型	int	2(16位)	$-2^{15} \sim 2^{15} - 1$ (-32 768~32 767)
短整型	short	2(16位)	$-2^{15} \sim 2^{15} - 1$ (-32 768~32 767)
长整型	long	4(32位)	$-2^{31} \sim 2^{31} - 1$ (-2 147 483 648~2 147 483 647)
无符号整型	unsigned int	2(16位)	$0 \sim 2^{16} - 1$ (0~65 535)

续表

类 型	类 型 名	字 节	取值范围
无符号短整型	unsigned short	2(16 位)	$0 \sim 2^{16} - 1$ (0~65 535)
无符号长整型	unsigned long	4(32 位)	$0 \sim 2^{32} - 1$ (0~4 294 967 295)

不同的编译系统或计算机系统对这几类整型数所占用的字节数有不同的规定。C 标准没有具体规定各类整数类型所占用的字节数,只要求 long 型数据长度不短于 int 型,short 型不长于 int 型。具体如何实现,由各计算机系统自行决定。例如,Turbo C 中的 short、int 都是 16 位,long 是 32 位;而 Visual C++ 中的 int、long 都是 32 位,short 是 16 位。

2. 整型常量

值为整数的常量称为整型常量,简称为整常量或整数。它包括正整数、负整数和零。C 语言中的整型常量有以下 3 种表示形式。

(1)十进制:就是通常意义下的十进制整常数。例如,123、-456、0。

(2)八进制:以 0 开头,后面跟几位数字(由数字 0~7 组成)。例如, $0123 = (123)_8 = (83)_{10}$; $-011 = (-11)_8 = (-9)_{10}$ 。

(3)十六进制:以 0x 或 0X 开头,后面跟几位数字(数字由 0~9、A~F 组成)。例如, $(0x123)_{16} = (291)_{10}$, $(-0x12)_{16} = (-18)_{10}$ 。

整型常量有短整型、基本整型、长整型和无符号型等不同类型。整型常量后可以用 u 或 U 明确说明数据为无符号整数,用 l 或 L 明确说明为长整型数。例如,011L 表示长整型的八进制数,代表十进制数 9,在计算机中占 4 字节;011 表示整型的八进制数,也代表十进制数 9,但在计算机中占 2 字节。

3. 整型变量

整型变量可以分为短整型、基本整型、长整型和无符号型等不同类型。例如:

```
short s;      /* 定义 s 为短整型变量 */
int i,j,k;    /* 定义 i,j,k 为基本整型变量 */
long n;       /* 定义 n 为长整型变量 */
```

定义以上语句后,编译程序会为 s、i、j、k、n 分别开辟相应字节的空间,而没有在存储单元中存放任何值。此时变量中的值是无意义的,称变量值“无定义”。

C 语言规定,可以在定义变量的同时给变量赋初值,也称为初始化。例如:

```
int i,j,s=0; /* 定义 i,j,s 为整型变量,s 初始化为 0 */
```

2.2.2 实型数据

1. 实型数据的分类

C 语言中的实型数据有单精度(float)、双精度(double)和长双精度(long double)三类。有关规定如表 2-3 所示。



视频
整型变量

表 2-3 实型数据类型

类 型	类 型 名	字 节	有效数字	取值范围
单精度	float	4(32 位)	6~7	$10^{-37} \sim 10^{38}$
双精度	double	8(64 位)	15~16	$10^{-307} \sim 10^{308}$
长双精度	long double	10(80 位)	18~19	$10^{-4931} \sim 10^{4932}$

ANSI C 没有具体规定每种类型数据的长度、精度和数值范围,由 C 编译系统自定。表 2-3 列出的是微机上常用的 C 编译系统(如 Turbo C, Borland C)的情况。

注意 C 语言中实型与数学中实数的区别。数学中实数的值是无限的,而实数在计算机中使用有限的存储单元存储,所以值是有限的。另外,数学中的精度可以是任意的,甚至是无限的,而计算机只能以有限个有效位表示精度。程序设计者可根据值域和精度的实际需要,选用适当的实型类型。

2. 实型常量

实型常量又称为实数或浮点数。在 C 语言中,实型常量只能用十进制表示,有以下两种表示形式。

(1)十进制小数形式。由数字和小数点组成(必须有小数点),如 123.0、.123(只有小数位)、123.(只有整数位)、0.0。

(2)指数形式,又称科学表示法。这种表示形式包含数值部分和指数部分,数值部分的表示方法同十进制小数,指数部分是一个可正可负的整型数,这两部分用字母 e 或 E 连接起来。这种形式类似于数学中的指数形式。在数学中,一个数可以用幂的形式来表示,如 123×10^3 ,在 C 语言中,123e3、1.23E5 都是它的合法表示。使用指数形式表示很大或很小的数比较方便。

使用实型常量需要注意以下几点。

(1)字母 e 或 E 之前必须有数字,e 后面的指数必须为整数。例如,e3、2.1e3.5、.e3、e 都不是合法的指数形式。

(2)规范化的指数形式。这种形式要求在字母 e 或 E 之前的小数部分,小数点左边应当有且只有一位非 0 数字。例如,2.3478e2、3.0999E5、6.46832e12 都是规范化的指数形式。用指数形式输出实数时,都是按规范化的指数形式输出的。

(3)许多 C 编译系统将实型常量作为双精度实数来处理,这样可以保证较高的精度,缺点是运算速度降低。在实数的后面加字符 f 或 F,如 1.65f,使编译系统按单精度处理实数。

3. 实型变量

C 语言中的实型变量分为单精度型(float)、双精度型(double)和长双精度型(long double)。对于每一个实型变量也都应该先定义后使用,可以在定义变量的同时进行初始化。例如:

```
float a,b;          /* 定义 a,b 为单精度型变量 */
double x,y,z;      /* 定义 x,y,z 为双精度型变量 */
double t=1.0,s=0; /* 定义变量 t,s 为双精度型变量,同时进行了初始化 */
```

由于实型数据的存储是有误差的,使用时要注意以下几点。

- (1) 不要试图用一个实数精确表示一个大整数,因为浮点数是不精确的。
- (2) 实数一般不判断“相等”,而是判断接近或近似。
- (3) 避免直接将一个很大的实数与一个很小的实数相加或相减,否则会“丢失”小的数。
- (4) 分析数据,根据需要选择数据类型是单精度还是双精度。

2.2.3 字符型数据

在处理数据时,经常会遇到像姓名、性别和住址等具有文本特征的数据,这些数据属于字符型数据。

1. 字符常量

用单引号括起来的单个字符为字符常量。例如:

合法的字符常量:'a','A','5','0',' ','.'

不合法的字符常量:"a"(双引号不合法)、『abc'(超过1个字符不能用单引号)。

字符常量在内存中占1字节,存放的是字符的ASCII码值。字符常量'A'的值为65,字符常量'a'的值为97。

2. 转义字符常量

转义字符是以“\”开头的具有特殊含义的字符,这些字符常量也必须括在一对单引号内。例如, '\n' 代表回车换行。表2-4列出了C语言中的转义字符。

表 2-4 转义字符

字符形式	功 能	字符形式	功 能
\n	回车换行	\\	反斜杠字符\
\t	横向跳过若干格(Tab 键)	\'	单引号字符
\v	竖向跳格	\"	双引号字符
\r	回车符	\ddd	三位八进制数代表一个字符
\f	换页符	\xhh	两位十六进制数代表一个字符
\b	退格符(Backspace 键)	\0	空值

使用转义字符需要注意以下几点。

- (1) 转义字符常量,如 '\n'、'\101'、'\141' 只代表一个字符。
- (2) 反斜线后的八进制数可以不用0开头,如 '\101' 代表的就是字符常量'A'。
- (3) 反斜线后的十六进制数只能由小写字母x开头,不允许用大写字母X,也不能用0x开头,如 '\x41' 代表字符常量'A'。

3. 字符串常量

字符串常量是由双引号括起来的一串字符,如" How are you ?""CHINA""a"。

C语言规定:在每个字符串的结尾加一个字符串结束标志,以便系统据此判断字符串是否结束。C语言规定以 '\0'(ASCII码为0的字符)作为字符串结束标志。

注意字符常量与字符串常量的区别。例如,'a'字符型常量占1字节,而"a"字符串常量占2字节;"ab\n"是字符串常量,占4字节;" "代表一个空串,占1字节存放 '\0'。

注意：字符串只能是常量，C语言中没有字符串变量。

4. 可对字符量进行的运算

字符数据以 ASCII 码存储的形式与整数的存储形式类似，这使得字符型数据和整型数据之间可以通用(当作整型量)。例如：

```
'B'-'A'=66-65=1      'a'+1=97+1='b'
'A'+32=65+32=97='a'  'b'-32=98-32=66='B' /*大小写字母之间的转换*/
'9'-'0'=57-48=9     4+'0'=4+48=52='4' /*数字字符与整数值之间的转换*/
```

字符量还可以进行关系运算。例如，'A'<'a'，值为真。因为在 ASCII 码表中，'A' 的值为 65，小于 'a' 的值 97。

5. 字符型变量

字符型变量用来存放字符数据，同时只能存放一个字符。在 C 语言中，字符型变量用关键字 char 进行定义，在定义的同时也可以初始化。例如：

```
char c1,c2,c3;
char ch='A';
```

所有编译系统都规定以 1 字节来存放一个字符，或者说，一个字符型变量在内存中占 1 字节。当把字符放入字符型变量时，字符型变量中的值就是该字符的 ASCII 码值，这使得字符型数据和整型数据之间可以通用(当作整型量)。具体表现为如下几点。

- (1) 可以将整型数据赋值给字符型变量，也可以将字符数据赋值给整型变量。
- (2) 可以对字符数据进行算术运算，相当于对它们的 ASCII 码进行算术运算。
- (3) 一个字符数据既可以以字符形式输出，也可以以整数形式输出。

【例 2-2】 大小写字母的转换。

程序代码如下。

```
#include <stdio.h>
main()
{
    char c1,c2,c3;
    c1='a';
    c2='b';
    c1=c1-32;
    c2=c2-32;
    printf("\n%c,%c\n",c1,c2);
    printf("%d,%d\n",c1,c2);
}
```

程序运行结果如下。

```
A,B
65,66
```

程序的作用是将两个小写字母转换为大写字母。在 ASCII 码表中，小写字母比对应的大写字母的 ASCII 码大 32，所以直接减 32 实现从小写字母到大写字母的转换；第一个

printf 函数将字符以字符形式输出,第二个 printf 函数将字符以整数形式输出。

2.3 C 语言的运算符和表达式

C 语言的运算符非常丰富,除了提供一般高级语言的算术运算符、关系运算符和逻辑运算符外,还提供赋值运算符、位操作运算符、自增自减运算符等,甚至数组下标、函数调用都作为运算符。本节主要介绍算术运算符、赋值运算符、自增自减运算符和逗号运算符。

2.3.1 算术运算符和算术表达式

1. 算术运算符

常见的算术运算符包括+、-、*、/、%和正负号。它主要对数值型数据进行一般的算术运算,其运算规则、运算对象、结合性如表 2-5 所示。

表 2-5 算术运算符及其运算规则

对象个数	名称	运算符	运算规则	运算对象类型	结果类型	结合性
单目	正	+	取正值	整型或实型	整型或实型	自右向左
	负	-	取负值			
双目	加	+	加法	整型或实型	整型或实型	自左向右
	减	-	减法			
	乘	*	乘法			
	除	/	除法			
	模(求余)	%	整数取余	整型	整型	

有关算术运算符需要说明以下几点。

(1)除法运算符/,如果两个整数相除,那么结果为整数,如 5/3 的结果为 1,舍去小数部分。如果两个运算对象中至少有一个是实型,那么结果就是实型。

(2)如果参加+、-、*、/运算的两个数有一个为实数,则结果为 double 型,因为所有实数都按 double 型进行计算。

(3)求余运算符%,要求两个操作数均为整型,结果为两数相除所得的余数。求余也称为求模。一般余数的符号与被除数符号相同。例如,8%5=3,-8%5=-3,8%-5=3。

(4)双目运算符优先级,*、/、%同级,+、-同级,并且前三个高于后两个。

2. 算术表达式

算术表达式是指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 语言语法规则的表达式。运算对象可以是常量、变量、函数等。例如,a * b/c-1.5+'a'。

需要注意的是,C 语言算术表达式与数学表达式的书写形式有一定的区别,具体如下。

(1)C 语言算术表达式的乘号(*)不能省略。例如,数学表达式“ $b^2 - 4ac$ ”对应的 C 表达式应该写成“b * b - 4 * a * c”。

(2)C 语言表达式中只能出现字符集允许的字符。例如,数学表达式“ πr^2 ”对应的 C 表

达式应该写成“PI * r * r”(其中 PI 是已经定义的符号常量)。

(3)C 语言算术表达式不允许有分子分母的形式。

(4)C 语言算术表达式只使用圆括号改变运算的优先顺序(不要使用{}、[])。可以使用多层圆括号,左右括号必须配对,运算时从内层括号开始,由内向外依次计算表达式的值。

3. 运算符的优先级与结合性

C 语言规定了进行表达式求值过程中各运算符的优先级和结合性。

(1)运算符的优先级。在表达式求值时,先按运算符的优先级别高低次序执行。例如,表达式 $a - b * c$ 等价于 $a - (b * c)$ ，“*”运算符优先级高于“-”运算符。

(2)运算符的结合性。如果在一个运算对象两侧的运算符的优先级别相同,则按规定的结合方向处理。有以下两种结合方向。

左结合性(自左向右结合方向):运算对象先与左面的运算符结合。

右结合性(自右向左结合方向):运算对象先与右面的运算符结合。

(3)在书写多个运算符的表达式时,应当注意各个运算符的优先级,确保表达式中的运算符能以正确的顺序参与运算。对于复杂表达式,为了清晰起见可以加圆括号“()”强制规定计算顺序。



视频
数学运算符的
优先级

2.3.2 赋值运算符和赋值表达式

在 C 语言中,可以通过赋值运算直接为变量提供数据。赋值运算是 C 程序中最为广泛的一种运算。

1. 赋值运算符

赋值运算符用赋值符号(=)表示,它的作用就是将一个数据赋给一个变量。

2. 赋值表达式

赋值表达式用于计算右边表达式的值,把右边表达式的值赋给左边的变量。其格式如下。

变量=表达式

例如,表达式 $x=y$ 的作用是将变量 y 存储单元中的数据赋给变量 x , x 中原有的数据被替换掉。赋值后,变量 y 中的内容不变。

需要说明以下几点。

(1)赋值运算符左边必须是变量,右边可以是常量、变量或常量、变量、函数调用组成的表达式。例如, $y=x+10$ 、 $y=func()$ 都是合法的赋值表达式,而 $a+b=c$ 是非法的。

(2)赋值符号“=”不同于数学的等号,它没有相等的含义,而是进行“赋予”操作。

例如,C 语言中 $x=x+1$ 是合法的(数学上不合法),它的含义是取出变量 x 的值加 1,再存放到变量 x 中。

(3)赋值运算时,当赋值运算符两边的数据类型不同时,将由系统自动进行类型转换。转换原则为先将赋值号右边表达式类型转换为左边变量的类型,然后赋值。

注意: 将实型数据(单、双精度)赋给整型变量时,将舍弃实数的小数部分。

(4)C 语言的赋值符号(=)除了表示一个赋值操作外,还是一个运算符,也就是说赋值运算符完成赋值操作后,整个赋值表达式还会产生一个所赋的值,这个值还可以利用。

例如,分析表达式 $x=y=3+5$ 。

$y=3+5$:先计算 $3+5$,将结果 8 赋值给 y , y 的值为 8, $y=3+5$ 整个赋值表达式值为 8。

$x=y=3+5$:将上面 $y=3+5$ 整个赋值表达式的值 8 赋给变量 x , x 的值为 8, $x=y=3+5$ 整个赋值表达式值为 8。

(5)赋值运算符的优先级只高于逗号表达式,比其他运算符的优先级都低,且具有自右向左的结合性。

3. 复合赋值表达式

复合赋值运算符由一个双目运算符和一个赋值运算符构成。C 语言规定可以使用 10 种复合赋值运算符,其中与算术有关的复合赋值运算符有: $+=$ 、 $-=$ 、 $*=$ 、 $/=$ 、 $\%=$ (注意:两个符号之间不可以有空格)。

例如, $n+=1$ 与 $n=n+1$ 等价; $n*=m+3$ 与 $n=n*(m+3)$ 等价。

$a+=a-=a+a$ 等价于 $a=a+(a=a-(a+a))$ 。若 a 为 9,则表达式的值为 -18。

采用复合赋值运算符一是为了简化程序,使程序简练,二是为了提高编译效率。

注意: 赋值运算符、复合赋值运算符的优先级比算术运算符低。

2.3.3 自增自减运算符和表达式



测试
等级考试真题

自增自减运算符是 C 语言中最具特色的两个单目运算符,其操作对象只有一个,这两个运算符既可以放在操作对象之前,也可以放在操作对象之后。它们的功能是自动将运算对象增 1 或减 1,然后把运算结果回存到运算对象中。例如:

(1) $++k$ 、 $--k$ 。

(2) $k++$ 、 $k--$ 。

上述两种形式在独立使用时,效果一样,都等价于 $k=k+1$ 或 $k=k-1$ 。区别在于,第(1)种形式是先使 k 增 1 或减 1,再使用 k ;第(2)种形式是先使用 k 的值,再将 k 增 1 或减 1。例如:

```
i=3;printf("%d",++i);
```

```
/* 输出 4 */
```

```
i=3;printf("%d",i++);
```

```
/* 输出 3 */
```

需要注意以下几点。

(1) $++i$ 、 $--i$ (前置运算):先自增、自减,再参与运算; $i++$ 、 $i--$ (后置运算):先参与运算,再自增、自减。

(2)自增、自减运算符只用于变量,而不能用于常量或表达式。例如, $6++$ 、 $(a+b)++$ 、 $(-i)++$ 都不合法。

(3) $++$ 、 $--$ 的结合方向是自右向左(与一般算术运算符不同)。例如,有表达式 $-i++$,其中 i 的值为 3。由于负号运算符与自加运算符的优先级相同,结果方向是自右往左,即相当于 $-(i++)$ 。此时 $++$ 属于后缀运算符,表达式的值为 -3, i 的值为 4。

(4)自增、自减运算符常用于循环语句中,使循环变量自动加 1,也用于指针变量,使指针指向下一个地址。

(5)C 语言中函数调用时,如果自增或自减出现在实参中,对于实参的求值顺序,C 标准

也无统一规定。例如,程序段

```
i=3;printf("%d,%d",i,i++);
```

在 Turbo C 环境下执行结果为 4,3,在 VC 环境下执行结果为 3,3。

当出现 $m+++n$ 这样的表达式时,C 编译系统在处理时尽可能多地(自左而右)将若干字符组成一个运算符,上式被理解成 $(m++)+n$,而不是 $m+(++n)$ 。

总之,尽量不要写别人看不懂(难看懂),也不知道系统会怎样执行的程序。

2.3.4 逗号运算符和逗号表达式

逗号运算符是 C 语言提供的又一种特殊的运算符。用逗号运算符将两个或多个表达式连接起来,构成一个完整的表达式,称为逗号表达式。

逗号表达式的一般形式如下。

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。

注意: 逗号表达式的优先级最低,结合顺序为自左向右。

例如,表达式“ $a=3*5,a++,a*4$ ”的值为 64,变量 a 的值为 16。

逗号表达式主要用于将若干表达式“串联”起来,表示一个顺序的操作(计算),在许多情况下,使用逗号表达式的目的只是想分别得到各个表达式的值,而并非一定需要得到和使用整个逗号表达式的值。

【例 2-3】 逗号表达式示例。

程序代码如下。

```
#include <stdio.h>
main()
{
    int x,a;
    x=(a=3,6*3);    /* a=3 x=18 */
    printf("%d,%d\n",a,x);
    x=a=3,6*a;      /* a=3 x=3 */
    printf("%d,%d\n",a,x);
}
```

程序的运行结果如下。

```
3,18
3,3
```

2.3.5 不同类型数据间的转换

1. 表达式计算中的数据类型转换

C 语言允许不同类型的数据混合运算,当不同类型的数据在运算符的作用下构成表达式时要进行类型转换,即把不同的类型先转换成统一的类型,然后进行计算。

通常数据之间的转换遵循的原则是“类型提升”,即如果一个运算符有两个不同类型的



测试
等级考试真题

数据操作,那么在运算之前,先将较低类型的数据提升为较高类型,然后进行运算,其结果数据是较高类型。一般来说,运算过程中系统自动转换的,称为隐式转换。

说明:

(1)转换按数据长度长的类型转换。例如,int型数据和long型数据运算时,先把int型转换为long型后再进行计算。

(2)所有浮点数运算都是以double型数据进行的。若一个表达式仅含float型的数据,也要先把float型转换为double型数据,再进行运算,结果自然是double型数据。

(3)char型数据和short型数据参与运算时,必须先将其转换为int型,结果为int型。



测试
等级考试真题

2. 强制类型转换表达式

强制转换是通过类型转换运算来实现的,用于把表达式的结果强制转换为类型说明符所表示的类型。其一般形式如下。

(类型说明符)表达式

例如:

```
(int)a           //将 a 的结果强制转换为整型量
(int)(x+y)       //将 x+y 的结果强制转换为整型量
(float)a+b       //将 a 的内容强制转换为浮点数,再与 b 相加。
```

说明:

(1)类型说明和表达式都需要加括号(单个变量可以不加括号)。

(2)无论隐式转换还是强制转换都是临时转换,不改变数据本身的类型和值。

【例 2-4】强制类型转换示例。

程序代码如下。

```
#include <stdio.h>
main()
{
    float f=5.75;
    printf("(int)f= %d\n", (int)f);
    printf("f= %f\n", f);
}
```

程序的运行结果如下。

```
(int)f=5
f=5.750000
```

等级考试重难点讲解与真题解析

本模块主要考查基本数据类型、运算符和表达式。通过对历年考试内容的分析,本模块属于重点考查内容。考点基本都属于基础知识,相对来说比较简单。对于自加、自减运算需要加强掌握,经常结合其他知识点综合考查。

一、重点、难点解析

1. 标识符

用户标识符只能由字母、数字和下划线组成,且第一个字符必须为字母或下划线。标识符的命名应该规范化,具体要根据下面几个原则来命名。

(1)选取有实际意义的标识符名称。

(2)标识符的名字不宜太长。必要时可以用一些缩写,有大小写之分。

(3)为了便于区分,不同的标识符不要取过于相似的名字。

(4)为了使说明部分阅读起来更加清晰,在对其进行类型说明时,应按照某种顺序对各种类型的变量进行集中说明,如先说明简单类型,再说明记录类型;在使用一个说明语句对同一类型的多个变量进行说明时,按照变量名中的字母顺序进行排列。

2. 常量

1) 整型常量

在C语言中,整型常量包括十进制整数、八进制整数(在八进制整数的前面加数字0)、十六进制整数(在十六进制整数的前面加0x)、长整型常量(在整数的后面加一个大写字母L或小写字母l)。

2) 字符常量

C语言的字符常量代表ASCII码字符集里的一个字符,在程序中要用单引号括起来。转义字符是以“\”开始的一个字符序列,主要用来表示一些不可显示的字符或控制代码,在程序中使用这种常量时也要括在一对单引号中。例如, '\ddd' 是1~3位八进制数ddd对应的符号, '\xhh' 是1~2位十六进制数hh对应的符号。

在内存中,字符数据以ASCII码存储,它的存储形式与整数的存储形式类似,因此,一个字符数据既可以以字符形式处理、输出,也可以以整数形式处理、输出。

3) 符号常量

符号常量的定义实际是宏定义的特例,是以“#define”开头的编译预处理命令。

为了区别于一般的变量名,符号常量通常用大写字母。例如, NULL、EOF 是系统提供的符号常量,分别与0、-1等值,使用前必须在程序开头加上 #include <stdio.h> 编译预处理命令。

3. 变量

变量必须先定义再使用,当程序运行时,每个变量都要占用连续的若干字节,所占用的字节数由变量的数据类型确定。

C语言规定,变量可以是任何一种数据类型,如整型(int)、短整型(short)、长整型(long)、无符号整型(unsigned int)、单精度型(float)、双精度型(double)、字符型(char)等基本类型。对变量的定义可以放在函数之外,也可以放在函数体中或复合语句中。如果是放在函数体中或复合语句中,则必须集中放在最前面。

4. 运算符及表达式

C语言的运算符十分丰富,必须分析各类运算符的特征,从而记住它们的优先级和结合性。例如,所有的单目运算符的优先级均为第二级,结合性均为自右向左等。

1) 自增自减运算符

自增自减运算符的作用是使变量的值增 1 或减 1。例如, $i++$ 相当于 $i=i+1$; $--i$ 相当于 $i=i-1$ 。因此, 自增自减运算本身也是一种赋值运算。

(1) $++$ 和 $--$ 单目运算符, 运算对象可以是整型变量也可以是实型变量, 但不能是常量或表达式。

(2) 用自增自减运算符构成表达式时, 既可以以前缀形式出现, 也可以以后缀形式出现。无论前缀还是后缀, 对于变量本身来说都具有相同的效果, 但作为表达式来说却有着不同的值。例如, 若有“ $\text{int } i=3; x=++i;$ ”, i 和 x 以及表达式的值都为 4; 而执行“ $x=i++;$ ”后, x 和表达式的值为 3, 但 i 的值为 4。

2) 不同类型数据之间的混合运算和转换

当不同类型的数据混合运算时, 系统会自动统一类型后再做运算等处理。在赋值表达式中, 当赋值号(=)左右两边对象类型不一致时, 系统自动将右边对象的类型转换成左边对象的类型后, 再赋值给左边的对象; 在算术表达式中, 运算对象先按“就高不就低”的原则, 统一类型后再进行运算。

有些运算符对运算对象的类型有严格要求。例如, 整除求余运算符%, 要求左右两边的操作数均为整型, $2.0\%5$ 是错的, 若想将 2.0 转换成 2, 可以采用强制类型转换运算符来处理, 即改写成 $(\text{int})2.0\%5$ 。

3) 长度运算符

长度运算符是单目运算符, 其运算对象可以是任何数据类型符或变量, 运算结果是所测数据类型应占用的字节数。注意, 运算对象必须用圆括号。例如, $\text{sizeof}(\text{float})$ 的结果为 4。

二、真题解析

1. 请选出可用作 C 语言用户标识符的一组标识符()。[2017 年 9 月]

- | | | | |
|---------|----------|--------|--------|
| A. void | B. a3_b3 | C. For | D. 2a |
| define | _123 | -abc | DO |
| WORD | IF | Case | sizeof |

解析 用户标识符只能由字母、数字和下划线组成, 且第一个字符必须为字母或下划线, C 中“-abc”和 D 中“2a”都错误。关键字是 C 语言系统设置特定含义、专门用途的字符序列, 不能作为其他用途, A 中“void”和“define”不允许, 所以答案为 B。

2. 请选出合法的 C 语言赋值语句()。[2017 年 9 月]

- A. $a=b=58$ B. $i++;$ C. $a=58, b=58$ D. $k=\text{int}(a+b);$

解析 A 和 C 最后没有分号都不是语句; D 错误, 应该描述为“ $k=(\text{int})(a+b);$ ”, 所以答案为 B。

3. 设“ $\text{int } a=12$ ”, 则执行完语句“ $a+=a-=a*a$ ”后, a 的值是()。[2017 年 9 月]

- A. 552 B. 264 C. 144 D. -264

解析 “ $a+=a-=a*a$ ”等价于“ $a=a+(a=a-(a*a))$ ”, 先计算“ $a=a-(a*a)$ ”, 即“ $a=12-(12*12)$ ”, a 的值和表达式的值都为 -132, 再计算“ $a=-132+(-132)$ ”, 结果为 -264, 所以答案为 D。

4. 字符(char)型数据在微机内存中的存储形式是()。[2017年9月]

- A. 反码 B. 补码 C. EBCDIC 码 D. ASCII 码

解析 字符数据在内存中占1个字节,存放的是字符的ASCII码值。所以答案为D。

5. 假设在程序中 a,b,c 均被定义成整型,且已赋大于1的值,则下列能正确表示代数式 $\frac{1}{abc}$ 的表达式是()。[2017年9月]

- A. $1/a * b * c$ B. $1/(a * b * c)$ C. $1/a/b/(float)c$ D. $1.0/a/b/c$

解析 第一个表达式分母漏掉括号,加括号后跟第二个表达式一样,因为 a,b,c 被定义成整型,且赋大于1的值,C规定两个整数相除结果为整数部分,这样前三个表达式结果都为0,所以答案为D。

6. 在C语言中,如果下面的变量都是 int 类型,则输出的结果是()。[2017年4月]

```
sum=pad=5; pAd=sum++,pAd++,++pAd;
```

```
printf("%d\n",pad);
```

- A. 7 B. 6 C. 5 D. 4

解析 第一个赋值语句使 pad 为 5,在C语言中标识符都有大小写之分,所以后面的语句都是针对另一个变量 pAd 操作的,没有改变 pad 的值,答案为C。

7. 设有语句“char a='\72';”,则变量 a()。[2017年4月]

- A. 包含1个字符 B. 包含2个字符 C. 包含3个字符 D. 说明不合法

解析 反斜线后跟一个八进制数来表示一个ASCII字符,所以答案为A。

8. sizeof(double)是()。[2017年4月]

- A. 一种函数调用 B. 一个双精度型表达式
C. 一个整型表达式 D. 一个不合法的表达式

解析 sizeof 是长度运算符,运算结果是所测数据类型应占用的字节数,答案为C。

9. 下面程序的输出是()。[2016年9月]

- A. 17 B. 18 C. 23 D. 24

```
main()
{
    int x=023;
    printf("%d\n",--x);
}
```

解析 x 初始化为八进制 023,十进制为 19,--x 使结果为 18,所以答案为 B。

10. 在C语言中,合法的长整型常数是()。[2015年9月]

- A. 0L B. 4962710 C. 0.054838743 D. 2.1869e10

解析 整数不带小数点,所以 C、D 是错误的,在数据后面加 l 或 L 说明为长整型数,所以答案为 A。

思考与练习

一、选择题

- C语言中基本的数据类型包括()。
 - 整型、实型、逻辑型
 - 整型、实型、字符型
 - 整型、字符型、逻辑型
 - 整型、实型、逻辑型、字符型
- 以下选项中,不正确的浮点型常量是()。
 - 160.
 - 0.12
 - 2e4.2
 - 0.0
- 以下选项中,不正确的字符型常量是()。
 - 'a'
 - '\x41'
 - '\101'
 - "a"
- C源程序中不能表示的进制是()。
 - 二进制
 - 八进制
 - 十进制
 - 十六进制
- 以下选项中不合法的用户标识符是()。
 - _1
 - AaBc
 - a_b
 - a—b
- C语言中运算对象必须是整型的运算符是()。
 - %
 - /
 - !
 - *
- 若变量已正确定义并赋值,符合C语言语法的表达式是()。
 - a=a+7;
 - a=7+b+c,a++
 - int(12.3%4)
 - a=a+7=c+b
- 不合法的八进制数是()。
 - 019
 - 0
 - 01
 - 067
- 表达式“18/4 * sqrt(4.0)/8”值的数据类型是()。
 - int
 - float
 - double
 - 不确定
- 以下选项中,能用作数据常量的是()。
 - o115
 - 0118
 - 1.5e1.5
 - 115L

二、编程题

- 设圆的半径为2.5,编程求该圆的周长和面积。
- 编写程序,若 $x=1\ 500$, $y=350$,求出它们的商数和余数并进行输出。
- 编写程序,设 $a=2$, $b=3$, $x=3.5$, $y=2.6$ 。计算下面表达式 z 的值。

$$z = \frac{a+b}{2} + \sqrt{a} - (y-x)$$

- 编写程序,将大写字母E转换为小写字母e,求e后的第5个字符并输出。

3

模块 3

顺序结构程序设计

案例引入——学生成绩管理中数据的输入、计算与输出



任务描述

学生成绩表参见表 2-1。分析成绩表中的各列数据,对每列数据使用适当类型的常量或定义适当的变量。

1. 完成以下任务

- (1)通过键盘输入某学生的任务成绩(t_1, t_2, t_3, t_4)、考勤(sick, l_e(late/early), absent)和期末成绩(f_e(final exam))的值。
- (2)使用变量计算某学生形成性考核成绩(avg)、终结性考核成绩(score)。
- (3)使用常量输出某学生的姓名(name),并输出任务平均分和终结性考核成绩(score)。
- (4)完成 2 位同学的信息输入,按照要求进行计算并输出结果。

2. 几点说明与要求

- (1)形成性考核成绩(avg)为 t_1, t_2, t_3, t_4 的平均值。
 - (2)终结性考核成绩(score)为 avg 的 40%+期末成绩(f_e)的 60%-考勤扣分。
 - (3)考勤扣分规定为病假(sick)每次扣 2 分,迟到早退(l_e)每次扣 5 分,旷课(absent)每次扣 20 分。
 - (4)格式输出要求清晰、整齐。
 - (5)对于实型数据采用保留一位小数输出。
 - (6)通过重复以上操作可以处理若干位学生的成绩。
- 假设考核等级规定为 D 不及格(<60)、C 及格($60\sim 79$)、B 良好($80\sim 89$)、A 优秀($90\sim 100$),考核等级(grade)应怎么处理?



任务目标

- (1)能够熟练地根据数据处理需求使用合适的数据类型常量,定义合适的数据类型变量。

- (2)能够熟练地根据数据处理需求正确编写表达式。
- (3)能够使用赋值表达式计算处理,使用正确的格式输入/输出,能进行顺序结构程序设计。
- (4)培养程序设计人员耐心、细致、追求完美的基本素质。

源代码展示

```
#include <stdio.h>
main()
{
    int t1,t2,t3,t4,sick,l_e,absent,i=1;
    float avg,f_e,score;
    printf("\n 请输入第 %d 位同学任务成绩(t1,t2,t3,t4)、考勤成绩(sick,l_e,absent)和期末成绩(f_e):\n",i);
    scanf("%d%d%d%d%d%d%d%f",&t1,&t2,&t3,&t4,&sick,&l_e,&absent,&f_e);
    avg=(float)(t1+t2+t3+t4)/4; /* 计算 avg */
    score=avg*0.4+f_e*0.6-(sick*2+l_e*5+absent*20); /* 计算 score */
    printf("姓名 任务平均分 综合成绩\n");
    printf("刘延运 %10.1f %10.1f\n",avg,score); /* 输出学生的姓名及成绩,姓名用字符串常量直接输出,以后可以用字符数组等实现 */
    i++;
    printf("\n 请输入第 %d 位同学任务成绩(t1,t2,t3,t4)、考勤成绩(sick,l_e,absent)和期末成绩(f_e):\n",i);
    scanf("%d%d%d%d%d%d%d%f",&t1,&t2,&t3,&t4,&sick,&l_e,&absent,&f_e);
    avg=(float)(t1+t2+t3+t4)/4;
    score=avg*0.4+f_e*0.6-(sick*2+l_e*5+absent*20);
    printf("姓名 任务平均分 综合成绩\n");
    printf("孙燕 %10.1f %10.1f\n",avg,score);
}
```

运行结果

程序运行结果如图 3-1 所示。

```
"D:\C\3\Debug\cjl2.exe"
请输入第1位同学任务成绩<t1,t2,t3,t4>、考勤成绩(sick,l_e,absent)和期末成绩(f_e):
80 60 80 60 0 0 0 96
姓名 任务平均分 综合成绩
刘延运          70.0      85.6

请输入第2位同学任务成绩<t1,t2,t3,t4>、考勤成绩(sick,l_e,absent)和期末成绩(f_e):
95 95 90 80 0 1 0 86
姓名 任务平均分 综合成绩
孙燕           90.0      82.6
Press any key to continue_
```

图 3-1 “学生成绩管理中数据的输入、计算与输出”程序运行结果

程序分析

分析成绩表中各个数据项,定义它们的数据类型,根据要求设计计算公式 $score = avg * 0.4 + f_e * 0.6 - (sick * 2 + l_e * 5 + absent * 20)$,输入某个学生的各项数据,然后进行计算,最后根据要求(保留一位小数)进行输出。其他学生的成绩按同样方法处理。

知识讲解

在进行程序设计时,通常采用3种程序结构,即顺序结构、选择结构和循环结构。其中顺序结构是最基本、最简单的程序结构,也是进行复杂程序设计的基础。赋值操作和输入/输出操作是顺序结构最典型的操作。通过本模块的学习,大家可以开始最简单的C程序设计。

3.1 程序的三种基本结构

为了提高程序设计的质量和效率,现在普遍采用结构化程序设计方法。结构化程序由顺序结构、选择结构和循环结构3种基本结构组成。

1. 顺序结构

顺序结构是最简单的一种程序结构,语句是按书写的顺序执行的,语句的执行顺序与书写顺序一致。顺序结构的传统流程图如图3-2所示,顺序结构的N-S流程图如图3-3所示。

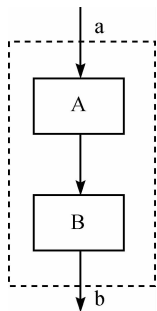


图 3-2 顺序结构的传统流程图

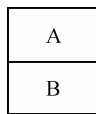


图 3-3 顺序结构的 N-S 流程图

2. 选择结构

选择结构又称分支结构,是根据给定的条件P是否成立而选择执行语句块A还是语句块B。选择结构的传统流程图如图3-4所示,选择结构的N-S流程图如图3-5所示。

3. 循环结构

循环结构是当满足某种循环条件时,将一条或多条语句重复执行若干遍,直到不满足循环条件为止。循环结构的传统流程图如图3-6所示,循环结构的N-S流程图如图3-7所示。

循环结构有以下两种类型。

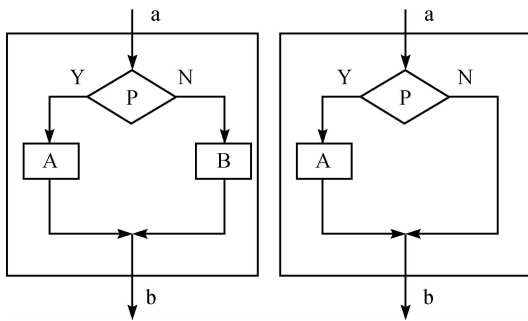


图 3-4 选择结构的传统流程图

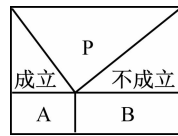
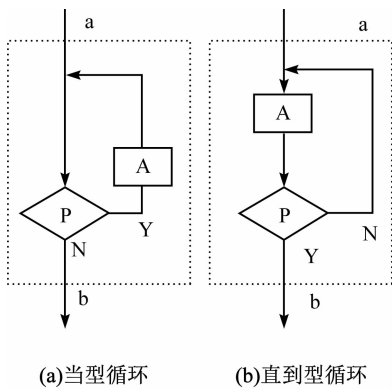


图 3-5 选择结构的 N-S 流程图

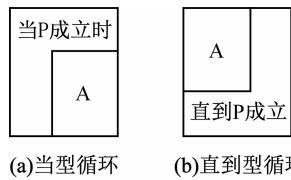
(1) 当型循环。当条件 P 成立时,反复执行 A 操作,直到 P 条件不成立为止。当型循环先判断,决定是否执行循环体,在条件 P 一次都不满足时,循环体 A 可能一次都不执行。

(2) 直到型循环。当条件 P 不成立时,反复执行 A 操作,直到 P 条件成立为止。直到型循环先执行循环体 A,然后判断条件 P,所以循环体至少执行一次。



(a) 当型循环 (b) 直到型循环

图 3-6 循环结构的传统流程图



(a) 当型循环 (b) 直到型循环

图 3-7 循环结构的 N-S 流程图

注意: 三种结构中的 A、B 框可以是一个简单的操作,也可以是 3 个基本结构之一,也就是说基本结构可以嵌套。

已经证明,由 3 种基本结构组成的算法可以解决任何复杂的问题。由 3 种基本结构组成的算法称为结构化算法,应当强调的是,在今后的程序设计中应当采用结构化程序设计方法。

3.2 赋值语句

赋值语句是由赋值表达式加分号构成的。例如,“ $a=b+c$ ”是赋值表达式,“ $a=b+c;$ ”则是赋值语句。

可以有如下几种赋值语句。

(1) 简单赋值语句,如“ $a=a+b;$ ”。

(2) 复合赋值语句,如“ $i+=1$;”等价于“ $i=i+1$;”。

(3) 自增自减赋值语句,如“ $i++$;”等价于“ $i=i+1$;”,“ $i--$;”等价于“ $i=i-1$;”。

(4) 逗号赋值语句,如“ $i=1,j=2$;”。

当同一个变量出现在赋值号两边时,对右边的变量进行的是取(读取)值运算,变量值不变;对左边的变量进行的是赋(写存)值运算,变量值更新。

例如,计数和累加运算的代码及过程如下。

```
i=0;s=0;           i           s
i=i+1;           1 /* 计数 */ scanf("%f",&cj); /* 输入成绩 */
s=s+i;           1 /* 累加 */ s=s+cj; /* 累加 */
i=i+1;           2           scanf("%f",&cj);
s=s+i;           3           s=s+cj;
i=i+1;           3           scanf("%f",&cj);
s=s+i;           6           s=s+cj;
i=i+1;           4           scanf("%f",&cj);
s=s+i;           10(1+2+3+4) s=s+cj; /* s=cj1+cj2+cj3+cj4 */
```

再例如,交换变量 x,y 的值的代码如下。

```
int x,y,t;
x=5;y=6;
t=x;x=y;y=t;
```



测试
等级考试真题

3.3 复合语句和空语句

复合语句、空语句和 C 语言中的其他语句一样,是程序设计中必不可少的语句,只有了解语句本身的功能和作用,并在程序中合理地使用,才能编写出清晰、高质量的程序。

1. 复合语句

用“{}”括起来一些语句(语句序列)称为复合语句,又称为语句块。

一般情况下,凡是允许出现语句的地方都允许使用复合语句。在程序结构上复合语句被看作一个整体的语句,但是内部可能完成了一系列工作。

需要注意的是,C 程序书写格式一般要求一条语句占一行,但在复合语句中,一般将彼此关联的或表示一个整体的一组较短的语句写在一行上。

在复合语句中,不仅可以有执行语句,也可以有定义部分,定义本复合语句中的局部变量。

例如:

```
{ int t;t=a;a=b;b=t;}
```

2. 空语句

空语句是只有一个分号的语句,它什么也不做。例如:

(1) 空循环 100 次。

```
for(i=0;i<100;i++)
; /*“;”表示循环体,表示一个延时或表示目前还不必做什么事情*/
```

(2)如果条件满足什么都不做,否则完成某些工作。

```
if()
; /*“;”表示 if 块,什么都不做*/
else
{ .....}
```

程序中有时需要加一个空语句来表示存在一条语句,但随意加分号也会导致逻辑上的错误,而且这种错误十分隐蔽,编译器也不会提示逻辑错误,初学者一定要小心、慎用。

3.4 数据的输入与输出

所谓输入与输出,是以计算机主机为主体而言的。从计算机向外部输出设备(如显示器、打印机、磁盘等)输出数据称为“输出”,从输入设备(如键盘、磁盘、扫描仪等)向计算机输入数据称为“输入”。

C语言本身不提供输入/输出语句,输入/输出操作是由函数来实现的。C语言的标准函数库中提供有许多用于实现输入/输出操作的库函数,使用这些函数时,只要在程序开始的位置上加上如下编译预处理命令即可。

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

或

```
#include "stdio.h"
```

它的作用是将输入/输出函数的头文件 `stdio.h` 包含到用户源文件中。其中, `h` 为 `head` 的缩写, `stdio` 是 `standard input & output` 的缩写。考虑到 `printf` 和 `scanf` 函数使用频繁,系统允许在使用这两个函数时可不加 `#include` 命令。

3.4.1 字符的输入与输出

1. 字符输出函数

字符输出函数 `putchar` 用于向输出设备(显示器)输出一个字符(可以是可显示的字符,也可以是控制字符或其他转义字符)。其语法格式如下。

```
putchar(字符型表达式);
```

例如:

```
putchar('y'); /* 输出字母 y */
```

```
putchar('\n'); /* 输出一个换行符 */
```

2. 字符输入函数

字符输入函数 `getchar` 用于从终端(键盘)输入一个字符,以 `Enter` 键确认。函数的返回值就是输入的字符。其语法格式如下。

```
getchar();
```



视频
字符输出函数

使用 getchar 函数需要注意以下两点。

(1)从键盘输入字符型常量时不用单引号,输入字符后,按 Enter 键。

(2)getchar()只能接受一个字符,getchar 函数得到的字符可以赋给一个字符变量或整型变量,也可以不给任何变量,作为表达式的一部分。

【例 3-1】 从键盘输入一个大写字母,要求以小写字母输出。

程序代码如下。

```
#include <stdio.h>
main()
{
    char c1,c2;
    c1=getchar();
    putchar(c1);
    putchar('\n');
    c2=c1+32;
    putchar(c2);
}
```

程序运行结果如下。

```
A
A
a
```

3.4.2 格式输出函数

1. 格式

格式输出函数 printf 有以下两种格式。

(1)直接输出字符串,其语法格式如下。

```
printf("字符串");
```

(2)计算若干任意类型表达式的值,按格式向输出设备输出。其语法格式如下。

```
printf("格式控制字符串",输出表列);
```

例如:

```
printf("%d,%c,%d",65,65,'A'+1);
/* 输出:65,A,66 */
```

对于格式(2)有以下两点说明。

①格式控制字符串。格式控制字符串可以是下列两种形式的组合。

格式说明:%格式字符。格式说明是用于规定对应输出项内容的输出格式,它由%和格式字符组成。

普通说明:原样输出的其他字符。一般使用“普通说明”给出输出结果的一些提示或间隔,使输出的值清晰、整齐。

②输出表列。输出表列是指要输出的数据,可以是常量、变量或表达式,输出多个数据时以“,”分隔。

2. 格式字符

对不同类型的数据用不同的格式字符。常用的有以下几种字符。

(1)d 格式符,用来输出十进制整数。

①%d:按整数数据的实际长度输出。

②%md:m为指定输出的宽度。若 $m \leq$ 数据的实际位数,按实际宽度输出;若 $m >$ 实际位数,则右对齐,左补空格。

③%-md:m为指定输出的宽度。若 $m \leq$ 实际位数,按实际宽度输出;若 $m >$ 数据的实际位数,左对齐,右补空格。

④%ld:输出长整型数据。

(2)o(是字母o不是零)格式符,用来输出八进制整数。%o表示将内存单元中的各位的值按八进制形式输出,输出的数值不带符号,即将符号位也一起作为八进制的一部分输出。

例如,有程序段如下。

```
int a=-1;
printf(" %d,%o",a,a);
/* 输出为:-1,177777 */
```

对长整型数据可以用“%lo”格式输出,同样也可以指定宽度。

(3)x 格式符,用来输出十六进制整数。%x表示以十六进制数形式输出整数,同样不会出现负的十六进制数。

例如,有程序段如下。

```
int a=-1;
printf(" %d,%o,%x",a,a,a);
/* 输出为:-1,177777,ffff */
```

同样可以用“%lx”输出长整型数,也可以指定输出的宽度。

(4)u 格式符,用来输出无符号整数。一个有符号整数(int)也可以用%u格式输出;反之,一个无符号数据也可以用%d格式输出。无符号整数也可以用%o或%x格式输出。

(5)c 格式符,用来输出一个字符。例如:

```
char ch='a';
printf(" %c",ch);
```

一个整数,只要它的值为0~255,也可以用字符形式输出。在输出前,系统会将该整数作为ASCII码转换为相应的字符;反之,一个字符数据也可以用整数输出。

【例 3-2】 整数与字符的输出示例。

程序代码如下。

```
#include <stdio.h>
main()
{
    char c='a';
    int i=97;
    printf(" %c,%d\n",c,c);
```



测试
等级考试真题


```
printf(" %c,%d\n",i,i);
}
```

程序运行结果如下。

```
a,97
```

```
a,97
```

(6)s 格式符,用来输出一个字符串。有以下几种用法。

①%s:输出字符串。例如:

```
printf(" %s","Hello");
```

输出“Hello”字符串(不包含双引号)。

②%ms:输出的字符串占 m 列。若字符串本身长度大于 m ,则按实际宽度输出;若串长小于 m ,则左补空格。

③%-ms:若串长小于 m ,字符串向左靠,右补空格。

④%m.n s:输出占 m 列,但只取字符串左边 n 个字符。这 n 个字符输出在 m 列的右侧,左补空格。

⑤%-m.n s:其中 m 、 n 含义同上, n 个字符输出在 m 列的左侧,右补空格。若 $n > m$,则 m 自动取 n 值,即保证 n 个字符输出。

(7)f 格式符,用来输出实数(float、double),以小数形式输出。有以下几种用法。

①%f:不指定字段的宽度,由系统自动设定,整数部分全部如数输出,并输出 6 位小数。单精度的有效位数一般为 7 位。

例如,有程序段如下。

```
float x,y;
x=111111.111;
y=222222.222;
printf(" %f",x+y);
/* 输出结果为:333333.328125 */
```

显然,只有前 7 位数字是有效数字。

双精度数也可以用%lf 格式输出,输出 6 位小数,有效数字一般是 16 位。

②%m.n f:指定输出的数据共占 m 列,其中 n 位小数。若 $m \leq$ 数据实际位数,按实际宽度输出;若 $m >$ 实际位数,则左补空格。

例如,有如下程序段。

```
float x=123.456;
printf(" %f, %10.2f, %.3f, %.0f",x,x,x,x);
/* 输出结果为:123.456001,□□□□123.46,123.456,123
```

③%-m.n f:指定输出的数据共占 m 列,其中 n 位小数。若 $m \leq$ 实际位数,按实际宽度输出;若 $m >$ 实际位数,则右补空格。

(8)e 格式符,以指数形式输出实数(float、double)。

①%e:不指定输出数据的宽度和小数位数。有的编译系统自动指定给出 6 位小数,指数部分占 5 位($e+005$)。数值按规范化指数形式输出,小数点前必须有且只有一位非零数字,所以数据共占 13 位。

② `%m.ne` 与 `%-m.ne`, `m,n` 和“-”含义与前相同。此处 n 指拟输出数据的小数部分(尾数)的小数位数。

(9) `g` 格式符。用来输出实数,它根据数据的大小,自动选 `f` 格式或 `e` 格式,且不输出无意义的零。

3. 使用 `printf` 函数的几点说明

(1) `printf` 的输出格式为自由格式,两个数之间是否有逗号、空格或回车,完全取决于格式控制,如果不注意,很容易造成数字连在一起,使得结果没有意义。例如:

```
printf("%d%d\n",x,y);
/* 若 x=123,y=345,则输出 123345 */
```

(2) 格式控制符必须含有与输出项一一对应的输出格式说明,类型必须匹配。若格式说明与输出项类型不一一匹配,则不能正确输出,而且编译时不会报错。若格式说明个数少于输出项个数,则多余的输出项不予输出;若格式说明个数多于输出项个数,则将输出一些毫无意义的数字乱码。

(3) 除了 X、E、G 外,其他格式字符必须用小写字母,如 `%d` 不能写成 `%D`。

(4) 在格式控制中,除了前面要求的输出格式,还可以包含任意合法字符,这些字符输出时将“原样照印”。还可以在“格式控制”字符串中包含转义字符,如“`\n`”。例如:

```
printf("a=%d,b=%d\n",a,b);
```

(5) 如果要输出“`%`”符号,可以在格式控制中用“`%%`”表示,则输出一个 `%`。

(6) 尽量不要在输出语句中改变输出变量的值,因为可能会造成输出结果的不确定性。例如,有如下程序段。

```
int k=8;
printf("%d,%d\n",k,++k);
```

输出结果为 9,9,而不是 8,9。这是因为调用函数 `printf` 时,参数是从右往左进行处理的。

3.4.3 格式输入函数

1. 格式

格式输入函数 `scanf` 用于按格式控制将输入设备的数据存入各变量。其语法格式如下。

```
scanf("格式控制字符串",地址表列);
```

(1) 格式控制字符串(与格式输出中的含义相同)。格式控制字符串可以是下列两种形式的组合。

① 格式说明: `%` 格式字符。格式说明用于规定对应输入项的输入格式,它由 `%` 和格式字符组成。

② 普通说明:原样输入的其他字符。有时也使用“普通说明”给出输入数据的间隔,如“`,`”。一般不用“普通说明”进行提示,必要的提示一般在 `scanf` 之前使用 `printf` 实现。

(2) 地址表列。地址表列是由若干个等待输入的内存单元地址组成的,地址项之间用逗号分隔。C 语言中变量地址的表示是在变量前加前缀符号“`&`”,形式如下。

&变量 1,&变量 2,...,&变量 n

2. 格式字符

%c:输入一个字符,按 Enter 键结束。

%d(ld):输入十进制整型数 int(long)。

%f(lf):输入 float(double)型数据。

%e(le):输入指数形式的 float 或 double 型数据。

%s:输入字符串,空白字符结束。

3. 使用 scanf 函数应当注意的问题

(1)scanf 函数中“格式控制”后面应当是变量地址,而不应当是变量名。若仅仅给出变量名会出错,但不报错,只给出警告。例如:

```
scanf("%d,%d",a,b);/* 不合法 */
```

(2)格式控制符必须含有与输入项一一对应的格式说明符,类型必须匹配;若格式说明与输入项类型不一一对应匹配,则不能正确输入,而且编译时不会报错。若格式说明个数少于输入项个数,scanf 函数结束输入,则多余的输入项将无法得到正确的输入值;若格式说明个数多于输入项个数,scanf 函数也结束输入,多余的数据作废,不会作为下一个输入语句的数据。

(3)如果在“格式控制”字符串中除了格式说明以外还有其他字符,则在输入数据时在对应位置应当输入与这些字符相同的字符。建议不要使用其他字符。例如:

```
scanf("%d,%d,%d",&a,&b,&c);/* 应当输入 3,4,5,不能输入 3 4 5 */
```

```
scanf("%d;%d;%d",&h,&m,&s);/* 应当输入 12;23;36 */
```

```
scanf("x,y,z=%d%d%d",&x,&y,&z);/* 应当输入 x,y,z=10 20 30 */
```

但使用者一般不了解设计者设置的输入格式串,这种情况一般先使用 printf 进行一定的提示。例如:

```
printf("Please input x,y,z:");
```

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

(4)在用%c格式输入字符时,空格字符和转义字符都作为有效字符输入。因为空格也是一个字符,所以不要用空格作为两个字符的间隔。

(5)在输入数据时,遇到下面情况认为数据输入结束。

①遇到空格或按 Enter 键或 Tab 键。

```
int a,b,c;
```

```
scanf("%d%d%d",&a,&b,&c);
```

```
/* 输入:12 34 (Tab) 567 并按 Enter 键后,a=12,b=34,c=567 */
```

②按指定的宽度结束。例如,%3d,只取3列。

③遇到非法的输入。例如:

```
float a,c;
```

```
char b;
```

```
scanf("%d%c%f",&a,&b,&c);
```

```
/* 输入:1234a123o.26 并按 Enter 键后,a=1234.0,b='a',c=123.0(而不是希望的 1230.26) */
```



测试
等级考试真题

(6)在 Visual C++ 环境下,要输入 double 型数据,格式控制符必须用%lf(或%le),否则数据不能正确输入。

(7)在 scanf 函数的格式字符前可以加入一个正整数指定输入数据所占的宽度,但不可以对实数指定小数位的宽度,如“scanf(“%5.2f”,&a);”是非法的。

(8)使用“*”跳过某个输入数据。可以在%和格式字符之间加入“*”,跳过对应的输入数据。例如:

```
int a1,a2,a3;
scanf(" %d% *d%d%d",&a1,&a2,&a3);
printf(" %d, %d, %d\n",a1,a2,a3);
/* 若输入:10203040,则输出为:10,30,40 */
```

3.5 技能训练

本模块介绍了结构化程序设计的3种基本结构及C语言实现输入/输出操作的函数。C语言输入/输出格式的规定比较烦琐,用得不对就不会得到预期的结果,而输入/输出又是最基本的操作,不少编程人员由于掌握不好而浪费大量的调试程序时间。但对于输入/输出格式没有必要死记硬背,要重点掌握常用的一些使用规则,其他部分可在需要时随时查阅。

【例 3-3】 从键盘输入3个整数,输出这3个数并计算其平均值。

程序代码如下。

```
#include <stdio.h>
main()
{
    int a,b,c;
    float average;
    printf("please input a,b,c:");
    scanf(" %d%d%d",&a,&b,&c);
    printf("a= %d,b= %d,c= %d\n",a,b,c);
    average=(float)(a+b+c)/3;
    printf("average= %.2f\n",average);
}
```

程序运行结果如下。

```
please input a,b,c:78 66 92
a=78,b=66,c=92
average=78.67
```

【例 3-4】 编写程序,已知苹果每斤 3.5 元,香蕉每斤 2.6 元,橘子每斤 1.8 元,要求输入各类水果的重量,输出应付钱数,再输入顾客付款数,打印出找顾客的钱数。

程序代码如下。

```
#include <stdio.h>
main()
{
    float w1,w2,w3,apple=3.5,banana=2.6,orange=1.8,price,money;
    int i;
    printf("请依次输入顾客购买苹果、香蕉和橘子的重量(不买的水果请输入0): \n");
    scanf("%f%f%f",&w1,&w2,&w3);
    price=apple*w1+banana*w2+orange*w3;
    printf("应付钱 %.2f 元\n",price);
    printf("请输入顾客付款数:");
    scanf("%f",&money);
    printf("应找钱数: %.2f 元\n",money-price);
}
```

程序运行结果如下。

请依次输入顾客购买苹果、香蕉和橘子的重量(不买的水果请输入0):

6 8 2

应付钱 45.40 元

请输入顾客付款数:100

应找钱数:54.60 元

【例 3-5】 输入三角形的三边长,求三角形面积(设输入的三边长 a, b, c 能构成三角形)。

程序代码如下。

```
#include <stdio.h>
#include <math.h> /* 调用数学函数库中的函数 */
main()
{
    float a,b,c,s,area;
    scanf("%f%f%f",&a,&b,&c);
    s=(a+b+c)/2;
    area=sqrt(s*(s-a)*(s-b)*(s-c)); /* sqrt 是求平方根的函数 */
    printf("a= %.2f,b= %.2f,c= %.2f\n",a,b,c);
    printf("area= %.3f\n",area);
}
```

程序运行结果如下。

12 8 16

a=12.00,b=8.00,c=16.00

area=46.476

等级考试重难点讲解与真题解析

本模块主要考查输入/输出和顺序结构设计方法。顺序结构程序设计属于最基本的程序设计方法之一,掌握这部分内容对后续学习及程序阅读作用极大。通过对历年考试内容的分析,这部分内容属于基础考点。主要考查格式输入函数 scanf 和格式输出函数 printf,这两个知识点一般不单独考核,常与其他知识点综合起来进行考查。

一、重点、难点解析

1. 输出函数 printf

printf 函数是 C 语言提供的标准输出函数,调用输出函数构成输出语句,格式如下。

```
printf("格式控制字符串",输出表列);
```

使用输出函数时要注意以下几点。

(1)若一个整数很大,要用“%ld”格式控制符控制输出,若用“%d”,一旦该数据超出范围,就不能被正确输出。

(2)在用一个 printf 函数输出多个数据时,若“格式控制字符串”中除含有%开头的控制符外,还含有其他字符,输出时,这些字符原样输出;若不含其他字符,各输出数据会连续排列,要注意题目给出的输出格式。

(3)所有函数的参数处理顺序都是从右至左的,输出时仍按各参数原有次序显示在屏幕上。

2. 输入函数 scanf

scanf 函数是 C 语言提供的标准输入函数,调用输入函数构成输入语句,格式如下。

```
scanf("格式控制字符串",地址表列);
```

使用输入函数时,要注意以下几点。

(1)scanf 中要求给出变量地址,若仅仅给出变量名会出错,但不报错,只给出警告。

(2)在用一个 scanf 输入多个数据时,若“格式控制字符串”中除含有%开头的控制符外,还含有其他字符,输入时,这些字符必须原样输入,所以在 scanf 的“格式控制字符串”中一般不含控制符以外的其他字符。若要提示,建议在 scanf 之前调用 printf 函数实现字符串提示。

(3)输入字符数据时,注意空格等也是一个字符。

(4)scanf 函数中没有精度控制,如“scanf("%5.2f",&a);”是非法的。

二、真题解析

1. 以下程序的输出结果是()。[2017年4月]

```
#include <stdio.h>
main()
{
    int i=010,j=10;
```

```
printf("%d,%d\n",++i,j--);
}
```

A. 11,10 B. 9,10 C. 010,9 D. 10,9

解析 i 赋值八进制数 010, 十进制数 8, ++i 的值为 9, j-- 的值为 10, 答案为 B。

2. 以下程序的输出结果是()。[2017 年 4 月]

```
#include <stdio.h>
#include <math.h>
main()
{
    int a=1,b=4,c=2;
    float x=10.5,y=4.0,z;
    z=(a+b)/c+sqrt((double)y)*1.2/c+x;
    printf("%f\n",z);
}
```

A. 14.000000 B. 015.400000 C. 13.700000 D. 14.900000

解析 表达式 z, 先计算(a+b)/c, 结果为 2; sqrt((double)y) 是求 y 的平方根, 结果为 2; 再计算 2*1.2/c, 结果为 1.2; 最终 z 的值就是 2+1.2+10.5=13.7。所以答案为 C。

3. 以下程序的输出结果是()。[2017 年 4 月]

```
#include <stdio.h>
main()
{
    int a=2,c=5;
    printf("a=%d,b=%d\n",a,c);
}
```

A. a=%2,b=%5 B. a=2,b=5
C. a=%d,b=%d D. a=%d,b=%d

解析 因为两个“%%”输出一个“%”, 后面的 d 就原样输出了, 答案为 D。

4. 已知字母 A 的 ASCII 码为十进制的 65, 下面程序的输出是()。[2016 年 9 月]

A. 67,D B. B,C C. C,D D. 不确定的值

```
main()
{
    char ch1,ch2;
    ch1='A'+5-'3';
    ch2='A'+6-'3';
    printf("%d,%c\n",ch1,ch2);
}
```

解析 '5'-'3' 的值为 2, '6'-'3' 的值为 3, 所以 ch1 的值为 'c', ch2 的值为 'd'。ch1 以整数输出, ch2 以字符输出, 所以答案为 A。

5. 执行下面程序中的输出语句, a 的值是()。[2016 年 9 月]

- A. 65 B. 20 C. 15 D. 10

```
main()
{
    int a;
    printf("%d\n", (a=3*5, a*4, a+5));
}
```

解析 “(a=3*5, a*4, a+5)”是一个逗号表达式,结果是最后表达式“a+5”的值。第一个表达式“a=3*5”的值和a的值都为15,第二个表达式的值为60,表达式“a+5”的值为20,所以答案为B。

6. 请读程序:

```
#include <stdio.h>
main()
{
    int a; float b, c;
    scanf("%2d%3f%4f", &a, &b, &c);
    printf("\na= %d, b= %f, c= %f\n", a, b, c);
}
```

若运行时从键盘上输入9876543210<CR>(<CR>表示回车),则上面程序的输出结果是()。[2016年9月]

- A. a=98, b=765, c=4321 B. a=10, b=432, c=8765
C. a=98, b=765.000000, c=4321.000000 D. a=98, b=765.0, c=4321.0

解析 scanf中的格式说明符为“%2d%3f%4f”,所以a接收2个数据98, b接收3个数据765, c接收4个数据4321,执行printf输出的结果为C,答案为C。

7. 下面程序的输出是()。[2016年9月]

- A. 0 B. 1 C. 3 D. 不确定的值

```
main()
{
    int x=10, y=3;
    printf("%d\n", y=x/y);
}
```

解析 输出“y=x/y”的值,在C语言中两个整数相除结果取整数部分,答案为C。

8. 下面程序的输出是()。[2015年9月]

```
main()
{
    int k=11;
    printf("k= %d, k= %o, k= %x\n", k, k, k);
}
```

- A. k=11, k=12, k=11 B. k=11, k=13, k=13
C. k=11, k=013, k=0xb D. k=11, k=13, k=b

解析 十进制数 11 的八进制数为 13,十六进制数为 b,又 k 的值分别以十进制、八进制和十六进制输出,所以答案为 D。

9. 以下说法中正确的是()。[2013 年 9 月]

- A. #define 和 printf 都是 C 语句 B. #define 是 C 语句,而 printf 不是
C. printf 是 C 语句,但 #define 不是 D. #define 和 printf 都不是 C 语句

解析 C 语言用“#define”进行宏定义,宏定义是用标识符来代表一个字符串,C 编译系统在编译前将这些标识符替换成所定义的字符串。printf 是 C 语言定义的格式输出函数,所以答案为 D。

10. 以下关于结构化程序设计的叙述中,正确的是()。[2010 年 9 月]

- A. 一个结构化程序必须同时由顺序、分支、循环三种结构组成
B. 结构化程序使用 goto 语句会很便捷
C. 在 C 语言中,程序的模块化是利用函数实现的
D. 由三种基本结构构成的程序只能解决小规模的问题

解析 C 语言中有顺序、选择、循环三种基本结构,这三种基本结构组成的算法可以解决任何复杂的问题,但并不是所有的程序都要用到这三种结构。goto 语句称为无条件转向语句,滥用 goto 语句会使程序流程毫无规律,可读性变差,应尽量避免使用。在 C 语言中,程序的模块化是利用函数实现的,所以答案为 C。



思考与练习

一、选择题

1. C 程序的 3 种基本结构是()。

- A. 顺序结构、选择结构、循环结构 B. 循环结构、递归结构、分支结构
C. 顺序结构、嵌套结构、循环结构 D. 顺序结构、转移结构、循环结构

2. 以下选项中不是 C 语句的是()。

- A. ; B. {int i; i++; printf("%d\n",i);}
C. x=2,y=10 D. { ; }

3. 以下合法的赋值语句是()。

- A. --i; B. k=int(a+b); C. a=18,b=18 D. a=b=12

4. 若变量已正确说明为 int 类型,要给 a、b、c 输入数据,以下正确的输入语句是()。

- A. scanf("%d%d%d",a,b,c);
B. read(a,b,c);
C. scanf("%D%D%D",&a,%b,%c);
D. scanf("%d%d%d",&a,&b,&c);

5. 若变量已正确定义 x=3.26894,以下语句的输出结果是()。

printf("%f\n",(int)(x*1000+0.5)/(float)1000);

- A. 3.27 B. 3.269
C. 3.268 D. 输出格式说明与输出项不匹配,输出无定值

6. 若有以下定义,变量 z 的值是()。

```
int x=10,y=3,z;
z=(x%y,x/y);
```

A. 1 B. 0 C. 4 D. 3

7. 以下程序段的输出结果是()。

```
int a=0,b=0,c=0;
c=(a--=a-5),(a=b,b+3);
printf("%d,%d,%d\n",a,b,c);
```

A. 3,0,-10 B. 0,0,5 C. -10,3,-10 D. 3,0,3

8. 若从终端输入给变量 a 赋以 123.19,则正确的输入语句是()。

A. scanf("%f",&a); B. scanf("%8.4f",&a);
C. scanf("%6.2f",&a); D. scanf("%f",&a);

9. 若变量已正确定义,要将 a 和 b 中的数进行交换,下面不正确的语句组是()。

A. $a=a+b,b=a-b,a=a-b$; B. $t=a,a=b,b=t$;
C. $a=t;t=b;b=a$; D. $t=b;b=a;a=t$;

10. 以下程序的输出结果是()。

```
main()
{
    int i=010,j=10,k=0x10;
    printf("%d,%d,%d\n",i,j,k);
}
```

A. 8,10,16 B. 8,10,10 C. 10,10,10 D. 10,10,16

二、编程题

1. 某工种按小时计算工资,每月劳动时间(小时) * 每小时工资 = 总工资,总工资中扣除 10% 公积金后为应发工资。编写程序,从键盘上输入劳动时间和每小时工资,输出应发工资。

2. 编程,输入一个华氏温度,要求输出摄氏温度。公式为: $c = \frac{5}{9}(F - 32)$ 。输出要求有文字说明,取 2 位小数。

3. 从键盘输入一个 3 位数整数,编写程序分别求其个位、十位、百位数并输出。