

项目 1

第一个 C 语言程序



工作任务

- 从微软官方网站下载 Visual Studio 2010 学习版。
- 在 Windows 7 或 Windows 10 环境下,安装 Visual Studio 2010 学习版。
- 编写、编译并运行第一个 C 语言程序。



项目要点

- 掌握集成开发环境的配置和使用。
- 掌握 C 语言程序的开发和调试过程。
- 了解软件编程规范。

任务 1 下载和安装 Visual C++ 2010

Visual C++ 2010 Express 是由微软提供的免费 C++ 开发环境,这个环境包括:源程序的输入和编辑,源程序的编译和连接,程序运行时的调试和跟踪,项目的自动管理,程序开发的各种工具,以及窗口管理和联机帮助等功能。

1. 获取和安装 Visual C++ 2010 Express

读者可以在微软开发者网站(<https://visualstudio.microsoft.com>)上下载 Visual C++ 2010 Express,如图 1-1 所示。



图 1-1 微软网站上的下载界面

网络安装版的安装程序大小仅为 3.2 MB, 双击即可进入安装向导, 如图 1-2 所示。



图 1-2 Visual C++ 2010 安装向导

单击“下一步”按钮并接受许可协议即可进入安装界面, 如图 1-3 所示。这里需要注意的是, 网络版安装程序的安装过程需要网络支持。



图 1-3 Visual C++ 2010 安装界面

(2) 在项目类型清单中,选择“Win32 控制台应用程序”项目,表示要生成一个 Windows 32 位控制台应用程序的项目。

(3) 在“位置”文本框中输入存放项目文件的文件夹路径。例如,将项目保存路径设置为 E:\C 语言程序设计\。

(4) 在“名称”文本框中输入项目名。例如,我们将要创建的项目是 helloworld,就在“名称”文本框输入 helloworld。

(5) 单击“确定”按钮,打开图 1-6 所示的对话框。选择附加选项中“空项目”复选框,单击“完成”按钮。



图 1-6 Win32 控制台应用程序的创建

执行完以上 5 个步骤后,系统自动给我们创建的项目文件加上文件扩展名“.dsp”。例如,系统在文件夹“E:\C 语言程序设计\”中产生一个项目文件 helloworld.dsp,如图 1-7 所示。

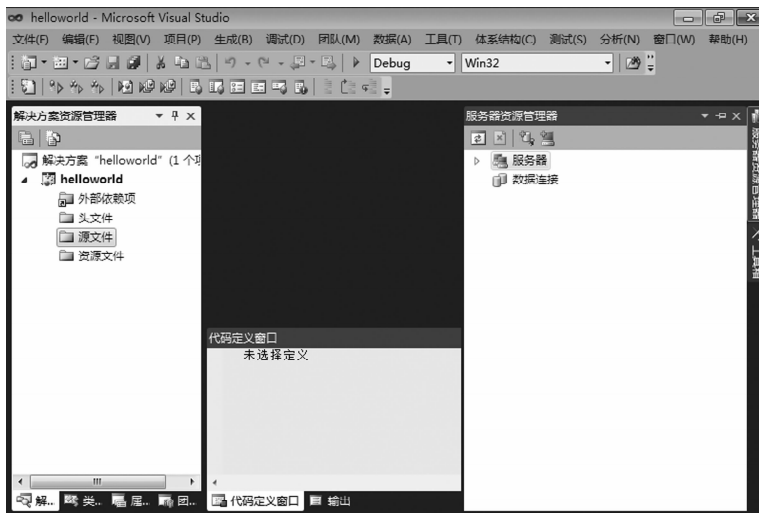


图 1-7 产生一个项目文件

2. 生成 C 源程序文件并将其加入项目文件

(1) 在“解决方案资源管理器”中,右击“源文件”,执行右键菜单中的“添加”→“新建项”

命令,在打开的“添加新项”对话框中选择 Visual C++模板下的“C++文件”,如图 1-8 所示。

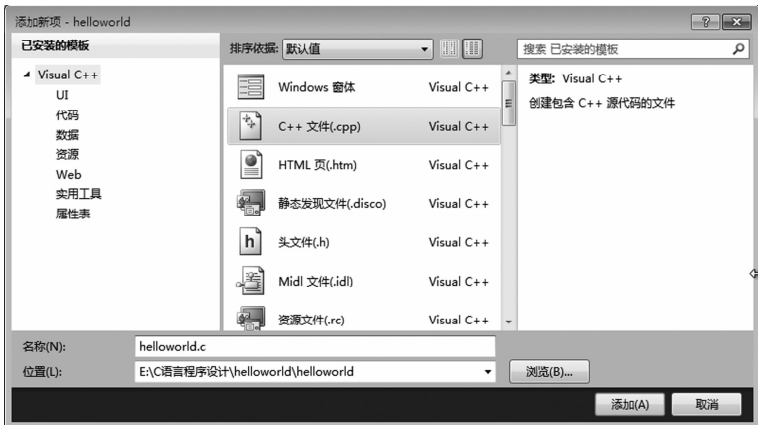


图 1-8 “添加新项”对话框

(2)在文件类型清单中选择“C++ Source File”选项,这里表示要生成一个 C 源程序。

(3)在“名称”文本框中输入 C 源程序文件名。注意,需要给源文件加上后缀名.c,否则系统会自动给文件加上.cpp 的后缀。例如,helloworld.c,如图 1-8 所示。

(4)单击“添加”按钮。这时就建立了一个新的 C 源程序文件,并已加入当前的项目文件中。进入图 1-9 所示的新建文件窗口就可以编写 C 语言程序了。

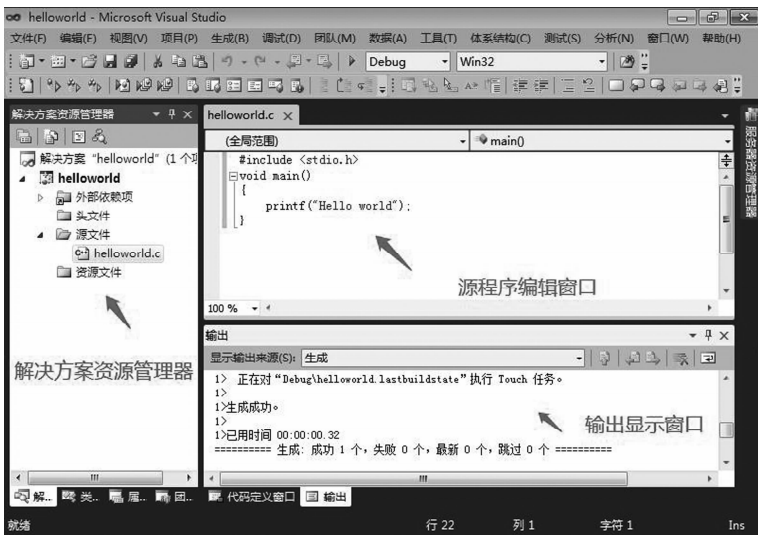


图 1-9 新建文件窗口

该窗口有三个窗格,左边的窗格为解决方案资源管理器;右边的窗格为源程序编辑窗格,用于输入或编辑源程序;下边的窗格为输出显示窗格,用来显示出错信息或调试程序的信息。

3. 输入和编辑 C 源程序

在源程序编辑窗格中输入以下程序:

```
#include <stdio.h>

void main()
{
    printf("Hello world!");
}
```

4. 保存 C 源程序文件

执行“文件”→“全部保存”命令,将源程序保存到相应的文件中。

5. 编译和连接源程序文件

执行“生成”→“生成解决方案”命令或按“F7”键,将源程序编译或编译连接,产生可执行文件。系统自动加上文件扩展名“.exe”,如“first.exe”。

在编译和连接期间,若出现错误,则在信息输出窗口给出错误或警告信息。改正错误后,重新编译或编译连接源程序,直到没有错误为止,如图 1-10 所示。

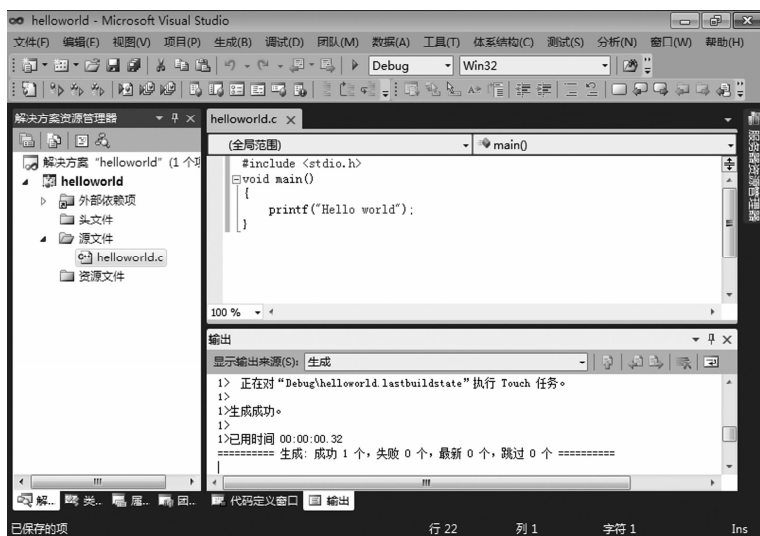


图 1-10 编译和连接源程序文件

6. 运行可执行文件

执行“调试”→“开始执行(不调试)”命令,或按“Ctrl+F5”快捷键,进入控制台窗口查看程序的运行结果,与 Windows 中运行 DOS 程序的窗口类似。图 1-11 所示为执行程序后弹出的 DOS 窗口中显示的程序执行结果。

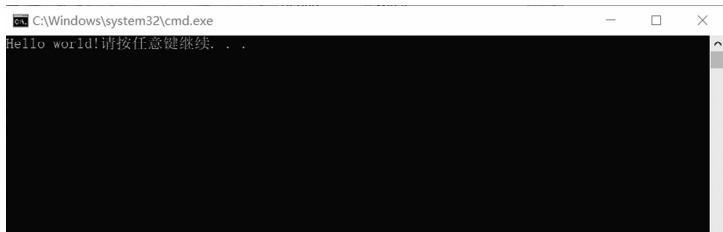


图 1-11 程序执行结果显示

7. 打开已存在的项目文件

可用以下两种方法打开已存在的项目文件：

(1) 执行“文件”→“打开”→“项目/解决方案”命令，然后在弹出的“打开项目”对话框中选择要打开的项目文件，如图 1-12 所示。

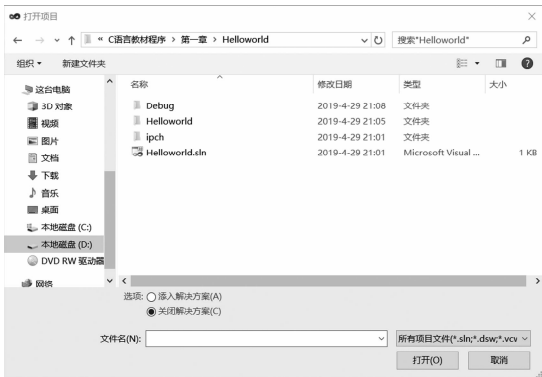


图 1-12 “打开项目”对话框

(2) 执行“文件”→“最近使用的项目和解决方案”命令，然后在弹出的子菜单中选择相应的项目文件。

注意：在调试一个应用程序时，Visual C++ 2010 集成开发环境一次只能打开一个项目文件。当一个程序调试完成后，要开始输入另一个程序时，必须先关闭当前的项目文件，然后为新课程建立一个新的项目文件。关闭当前的项目文件的方法是：执行“文件”→“关闭解决方案”命令。

8. 退出 Visual C++ 2010 集成开发环境

执行“文件”→“退出”命令，可以退出 Visual C++ 2010 集成开发环境。

注意：在建立 C 语言源文件时，也可以直接建立.c 文件，不需要先建立工程文件，再将.c 文件加入。

任务 3 小试牛刀

(1) 输入如下源程序，然后编译运行并观察结果。

```
#include <stdio.h>

void main()
{
    printf(" * \n");
    printf(" * * \n");
    printf(" * * * \n");
    printf(" * * * * \n");
    printf(" * * * * * \n");
}
```

(2) 输入如下源程序, 然后编译运行并观察结果。

```
#include <stdio.h>

void main()
{
    int a,b,sum;
    a=100;b=200;
    sum=a+b;
    printf("the sum is %d\n",sum);
}
```



应试宝典

【考点解析】

1. C 语言概述

C 语言是一种结构紧凑、使用方便、程序执行效率高的编程语言, 它包括 9 种控制语句、32 个关键字和 34 种运算符。C 语言的数据结构非常丰富, 它的多种数据类型可以实现链表、树、栈等复杂的运算, 并且用结构化控制语句(if...else、for 语句等)来实现函数的模块化。C 语言的语法不太严格, 程序设计自由度大, 用 C 语言编写的程序可以直接访问物理地址, 还可以直接对硬件进行操作。C 语言是一种移植性比较好的语言。

2. C 语言的构成

C 语言的源程序是由函数构成的, 每一个函数完成相对独立的功能, 其中, 每个源程序中必须有且只能有一个 main() 函数。C 程序总是从 main() 函数开始执行的。在函数后面用一对大括号“{}”括起来的部分称为函数体。C 语言规定每个语句以分号结束, 分号是语句组成部分不可缺少的一部分, 它在每条语句的最后出现。但预处理命令、函数头之后不能加分号。程序的注释部分应放在“/*”与“*/”之间, 注释部分允许出现在程序的任意位置。

程序中以“#”开头的语句称为预处理命令。

3. C程序的生成过程

C程序是先由源文件经编译生成目标文件,然后经过连接生成可执行文件。源文件的扩展名为.c,目标文件的扩展名为.obj,可执行文件的扩展名为.exe。

【真题全解】

1. C语言模块化是通过()来实现的。

- A. 变量 B. 函数 C. 程序行 D. 语句

【答案】B

【解析】C语言的源程序是由函数构成的,其模块化是通过函数来体现的,故选B。

2. 以下叙述中正确的是()。

- A. 书写源程序时,必须注意缩进格式,否则程序会有编译错误
B. 程序的主函数名除main外,也可以使用Main或_main
C. 程序可以包含多个主函数,但总是从第一个主函数处开始执行
D. 在C程序中,模块化主要是通过函数来实现的

【答案】D

【解析】C语言的源程序对缩进没有要求,所以A选项错误。C语言中区分大小写,因此main不能写成Main或_main,所以B选项错误;一个C程序有且只有一个主函数,所以C选项错误。

3. 以下叙述中正确的是()。

- A. C语言程序所调用的函数必须放在main函数的前面
B. C语言程序总是从最前面的函数开始执行
C. C语言程序中main函数必须放在程序的开始位置
D. C语言程序总是从main函数开始执行

【答案】D

【解析】C语言所调用的函数可以放在调用它的函数前,也可以放在调用它的函数后,所以A选项错误;C语言程序总是从main函数开始执行,所以B选项错误;main函数位置可以放在开始,也可以放在中间,也可以放在后面,所以C选项错误。

4. 以下叙述中正确的是()。

- A. 每个后缀为.c的C语言源程序都可以单独进行编译
B. 每个后缀为.c的C语言源程序都应该包含一个main函数
C. 在C语言程序中,main函数必须放在其他函数的最前面
D. 在C语言程序中,只有main函数才可单独进行编译

【答案】A

【解析】一个普通的C函数可以单独作为一个C的程序文件而包含到其他程序中,每个后缀为.c的C语言源程序都可以单独进行编译,所以B、D选项错误。main函数可以放在任

意位置,所以 C 选项错误。

5. 我们所写的每条 C 语句,经过编译最终都将转换成二进制的机器指令。关于转换,以下说法错误的是()。

- A. 一条 C 语句可能会被转换成零条机器指令
- B. 一条 C 语句可能会被转换成多条机器指令
- C. 一条 C 语句对应转换成一条机器指令
- D. 某种类型和格式的 C 语句被转换成机器指令的条数是固定的

【答案】C

【解析】一个 C 语句经过编译后产生若干条机器指令,C 选项中表述一条语句对应一条指令是错误的,故选 C。D 选项的说法是比较正确和全面的。

数据类型与表达式



工作任务

- 深入理解 C 语言的数据类型和表达式。
- 掌握简单 C 语言程序的编写和调试方法。
- 按要求完成程序的编写。



项目要点

- 掌握 C 语言数据类型和表达式的用法。
- 掌握 C 语言程序的编写和调试过程。

任务 1 编写打印销售单程序

小刘同学利用假期在一家超市打工,发现打印小票程序很有趣,他想利用 C 语言程序把购物小票打印出来。

小票的形式如下:

欢迎惠顾				
商品	单价	单位	数量	小计
可口可乐	2.50	听	10	25.00
冰红茶	3.00	瓶	10	30.00
饼干	15.00	盒	2	30.00
面包	5.00	袋	1	5.00
合计	90.00 (元)			

付款 100.00 元

找零 10.00 元

小刘同学想,我可以将每件商品的单价作为一个变量,用这个变量乘以数量就能得到小计,然后打印出每行内容,在打印的同时把合计金额记录下来,最后输入付款的金额,用付款的金额减去合计就是找零的钱数。我们已经学会了 printf 语句,利用这个语句就能输出每行的内容。

1. 编程思路

程序的功能包括按行打印和统计计算。按行打印可以使用 printf 函数,为了打印整齐,需要用到\t转义字符,这是一个制表符,表示在下一个输出区输出。另外,程序需要统计每笔商品的小计,使用单价 $\times$ 数量的公式,因此需要为每个商品声明一个变量表示单价和一个变量表示小计。同时声明一个 total 变量作为合计,每计算出一个 small 就使用+=复合赋值运算符进行累加。由于表示价格的变量都是两位小数,因此应该将其声明为 float 类型。

2. 程序代码

小刘同学编写的程序如下:

```
#include <stdio.h>
void main()
{
    printf("欢迎惠顾\n");
    printf("商品\t单价\t单位\t数量\t小计\n");
    float total=0;
    float cocla=2.5;float cocla_total=cocla*10;total+=cocla_total;
    printf("可口可乐\t%f\t听\t10\t%f\n",cocla,cocla_total);
    float redtea=3;float redtea_total=redtea*10;total+=redtea_total;
    printf("冰红茶\t%f\t瓶\t10\t%f\n",redtea,redtea_total);
    float cookie=15;float cookie_total=cookie*2;total+=cookie_total;
    printf("饼干\t%f\t盒\t2\t%f\n",cookie,cookie_total);
    float bread=5;float bread_total=bread*1;total+=bread_total;
    printf("面包\t%f\t代\t1\t%f\n",bread,bread_total);
    printf("合计\t\t\t%f\n",total);
    float cash;cash=100;
    printf("付款\t\t\t%f\n",cash);
    printf("找零\t\t\t%f\n",cash-total);
}
```

3. 运行结果

小刘同学的程序运行结果如图 2-1 所示。



图 2-1 购物小票打印结果

4. 改进与拓展

你能将小刘同学的程序在你的计算机中运行出来吗？另外，在图 2-1 中还有许多需要优化的问题，如单价的小数位过多，很多地方没有对齐，请你试着把这些问题优化一下。

另外，在金额打印上小数点后面出现了 6 个零，如何只打印出两位小数呢？我们可以将浮点打印格式字符从 %f 改为 %m.nf，表示指定数据占 m 列，其中有 n 位小数，如果数值长度小于 m ，左端用空格补齐。

大家在生活中一定遇到过很多超市小票，有的超市还会出现打折等情况，大家可以试试看。另外，校园卡充值、公交卡充值都会出现小票，大家也可以试试。比一比看看谁编写的程序最漂亮、功能最强大。

任务2 计算三角形面积

小张同学暑假在社区做义务老师，孩子们的暑假作业中有这样一道题：请同学们写出能够构成三角形的三个边的长度，并求出三角形的面积。孩子们的答案五花八门，这可急坏了小张同学，他想到可以编写一个程序，输入三个边长就能够计算出三角形的面积。

1. 编程思路

首先需要定义三个变量来存储三角形的三个边 a 、 b 、 c ，有些同学使用了小数，在声明时需要使用单精度实型。再定义一个变量 $area$ 用来存放三角形的面积。三角形的面积公式为

$$area = \sqrt{s(s-a)(s-b)(s-c)}$$

$$s = \frac{a+b+c}{2}$$

在 C 语言中根号用 $\text{sqrt}()$ 函数计算，这个函数在库函数 math.h 中进行了说明，因此需要将 math.h 头文件引用(include)进来。

2. 程序代码

```
#include <stdio.h>
#include <math.h>
void main()
{
    float a,b,c,area,s;
    printf("输入三角形的三个边");
    scanf("%f%f%f",&a,&b,&c);
    s=(a+b+c)/2;
    area=sqrt(s*(s-a)*(s-b)*(s-c));
    printf("三角形面积是 area= %f\n",area);
}
```

3. 运行结果

运行程序得到图 2-2 所示的结果。

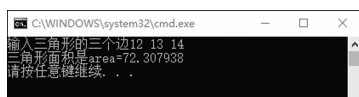


图 2-2 求三角形面积结果

4. 改进与拓展

通过这个程序,我们了解了 C 语言在数学方面的应用。在求三角形面积的程序中,当用户输入任意的三个边长时,可能出现两种情况:一种是能够构成三角形;另一种是无法构成三角形,也就是要满足三角形的任意两边之和大于第三边且任意两边之差小于第三边的要求。大家会发现,如果出现第二种情况结果会是 0.000000,如图 2-3 所示。因此,在计算之前可以进行判断,并给出一定的提示。在编写的程序中,一旦出现用户输入内容,一定要配合一定的代码对输入的信息进行检查并给出一些提示,这也是人机交互过程中重要的一部分。

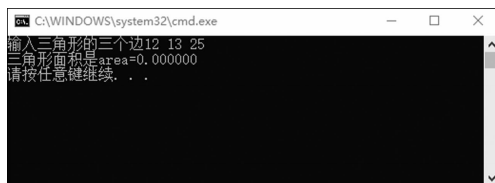


图 2-3 不能构成三角形的结果

另外,大家可以利用圆的公式,编写一个输入半径求圆的周长和面积的程序。或者求方程 $ax^2 + bx + c = 0$ 的根,输入 a 、 b 、 c 的值,计算出方程的根,这里需要注意判别方程根的情况。



【考点解析】

C 语言的数据类型包括数值型、字符型、布尔型等,其中数值型包括整型和实型;C 语言的表达式包括算数表达式、逗号表达式等。

1. 标识符

在 C 语言中,变量名、函数名、数组名等按照一定规则命名的符号称为标识符。

标识符的命名规则如下:

- (1)合法的标识符由数字、字母和下画线组成。
- (2)标识符的第一个字符必须是字母或下画线,不能是数字。
- (3)C 语言中要区分大小写。
- (4)标准 C 中没有限制标识符的长度,但有些编译系统规定标识符只有前 8 位有效。

C 语言的标识符分为以下三类:

- (1)关键字:C 语言规定了一些专用的标识符,它们有固定的含义,不能更改。
- (2)预定义标识符:这类标识符在 C 语言中和关键字一样也有特定的含义,这些类别有:库函数的名字,如 printf;预编译处理命令,如 define。
- (3)用户标识符:由用户根据题意定义的标识符就是用户标识符。一般是给变量、函数、数组或文件等命名,定义时应做到见名知意。

2. 常量与变量

在程序运行中,其值不能改变的量称为常量。常量有 5 种类型:整型常量、实型常量、字符型常量、字符串常量和符号常量。

变量就是值可以改变的量。变量要有变量名,在内存中占据一定的存储单元,存储单元中存放的是该变量的值。不同类型的变量,其存储单元的大小不同,变量在使用前必须先定义。

(1)整型变量。整型变量分为 4 种:基本型(int)、短整型(short int 或 short)、长整型(long int 或 long)和无符号型(unsigned int, unsigned short, unsigned long)。

(2)实型变量。实型变量分为单精度型(float)和双精度型(double)两种。在一般的系统中, float 型数据在内存中占 4 个字节(32 位), double 型数据占 8 个字节(64 位)。单精度实型提供了 7 位有效数字,双精度实型提供了 15~16 位有效数字。实型常量不分 float 型和 double 型,一个实型常量可以赋给一个 float 型或 double 型变量,但变量根据其类型截取实型中相应的有效数字。

(3)字符变量。字符变量用来存放字符常量,字符变量用关键字 char 定义,每个字符变量只能存放一个字符。

3. C 语言的运算符

(1)C 语言几乎把所有的基本操作都作为运算符来处理,如赋值、比较等,具体运算符见

表 2-1。

表 2-1 C 语言运算符

名称	运算符
算数运算符	+, -, *, /, %
关系运算符	>, <, =, ==, !=, <=, >=
位运算符	>>, <<, ~, &, , ^
逻辑运算符	!, , &&
条件运算符	?:
指针运算符	&, *
赋值运算符	=
逗号运算符	,
求字节数运算符	sizeof
强制类型转换运算符	(类型名), 如(int)

另外,还可以按照运算的对象个数把运算符分为单目运算符(如!)、双目运算符(如+、-)和三目运算符(如?:)。

(2)运算符的结合性和优先级。

①在 C 语言的运算符中,所有单目运算符、条件运算符、赋值运算符及其扩展运算符的结合方向都是从右到左,其余运算符的结合方向都是从左到右。

②各类运算符的优先级是:初等运算符>单目运算符>算术运算符(先乘除,后加减)>关系运算符>逻辑运算符(不包括!,! 按单目运算符执行)>条件运算符>赋值运算符>逗号运算符。

(3)逗号运算符和逗号表达式。逗号运算符的功能是将几个表达式连接起来,如“a=b+c,b=a,c=a+b;”称为逗号表达式。逗号表达式的运算过程是:从左往右依次求出每个表达式的值,最后一个表达式的值作为逗号表达式的值。

(4)自加、自减运算符。自加运算符++和自减运算符--的作用是使运算变量的值增 1 或减 1,它们都是单目运算。自加运算符和自减运算符的使用方法比较灵活,可以放在变量的前面,也可以放在变量的后面。放在前面时,如++i、--i,先执行自加或自减过程,再使用此时变量的值参与运算;放在后面时,如 i++、i--,先使用变量的值参与运算,再执行自加或自减的过程。

(5)复合赋值运算符。在 C 语言中,=称为赋值运算符,用来给一个变量赋值,由赋值运算符组成的表达式称为赋值表达式。

在赋值运算符之前加上其他运算符可以构成复合赋值运算符。其中,与算术运算有关的复合赋值运算符有+=、-=、*=、/=和%=等。例如,“a+=b”指的运算规则为 a=a+b。

【真题全解】

1. 以下选项中不属于 C 语言标识符的是()。

- A. 常量 B. 用户标识符 C. 关键字 D. 预定义标识符

【答案】A

【解析】常量是指在程序运行过程中其值不能改变的量,如 1、1.0、'a' 等。C 语言的标识符分为三类:关键字、预定义标识符和用户标识符。常量不属于标识符,故选 A。

2. 以下叙述中正确的是()。

- A. C 语言中的关键字不能作为变量名,但可以作为函数名
B. 标识符的长度不能任意长,最多只能包含 16 个字符
C. 标识符总是由字母、数字和下画线组成,且第一个字符不能为数字
D. 用户自定义的标识符必须见名知义,如果随意定义,就会出现编译错误

【答案】C

【解析】C 语言的关键字既不能作为变量名,也不能作为用户函数名,所以 A 选项错误。标识符的命名规则,即由下画线、字母、数字组成,且必须以下画线和字母开始,也不能与关键字一致。不同版本的 C 语言规定标识符的有效长度可能会不同,但没有限制最大长度,所以 B 选项错误。用户自定义的标识符只要满足标识符命名规则即可,并没有特殊要求,不一定见名知义,所以 D 选项错误。

3. 以下叙述中正确的是()。

- A. 只能在函数体内定义变量,其他地方不允许定义变量
B. 常量的类型不能从字面形式上区分,需要根据类型名来决定
C. 预定义的标识符是 C 语言关键字的一种,不能另做它用
D. 整型常量和实型常量都是数值型常量

【答案】D

【解析】变量可以定义在函数体外,也可以定义在函数体内,所以 A 选项错误。常量的类型可以从字面形式上区分,如 1 为整型常量,1.0 为实型常量,a 为字符型常量,所以 B 选项错误。预定义的标识符不是 C 语言的关键字,所以 C 选项错误。

4. 以下叙述中正确的是()。

- A. 在 C 程序中的八进制和十六进制可以是浮点数
B. 整型变量可以分为 int 型、short 型、long 型和 unsigned 型四种
C. 八进制数的开头要使用英文字母 o,否则不能与十进制区分开
D. 英文大写字母 X 和小写字母 x 都可以作为二进制数的开头字符

【答案】B

【解析】C 程序中的实型常量只有十进制一种数制,所以 A 选项错误。八进制数还可以 0 打头,所以 C 选项错误。C 语言中整数只有十进制、八进制、十六进制三种,没有二进制,所以 D 选项错误。

5. 以下选项中合法的常量是()。

A. 9 9 9

B. 2.7e

C. 0xab

D. 123E 0.2

【答案】C

【解析】A 选项中含有非法的空格,所以错误。C 语言中的十六进制常量以 0x 开头,所以 C 正确。实型常量中 e 的前后必须均有数据,且其后必须为整数,所以 B、D 选项错误。

6. 以下选项中非法的 C 语言字符常量是()。

A. '\007'

B. '\b'

C. 'aa'

D. '\xaa'

【答案】C

【解析】C 选项中的字符串常量应该用双引号表示,即"aa",故选 C。

7. 以下选项中非法的 C 语言字符常量是()。

A. '\x9d'

B. '9'

C. '\x09'

D. '\09'

【答案】D

【解析】本题考查字符常量,字符常量是用单引号引起来的一个字符。例如,'a'、'='、'+'等。转义字符是一种特殊的字符常量。转义字符以反斜杠\开头,后跟一个或几个字符。D 选项中,\0 表示后面的字符是八进制数,出现 9 是非法的,所以选择 D。

8. 以下选项中合法的字符常量是()。

A. "x"

B. 'X'

C. 'abc'

D. '\'

【答案】B

【解析】字符常量是用单引号引起来的一个字符。A 选项中,使用双引号标识字符串常量,所以是错误的;C 选项中,'abc' 是字符串,字符常量只能是单个字符,不能是字符串;D 选项中,以反斜线\开头的是转义字符。故选 B。

9. 以下叙述中正确的是()。

A. 转义字符要用双引号引起来,以便与普通的字符常量区分开

B. 字符常量在内存中占 2 个字节

C. 字符常量需要用单引号引起来

D. 字符常量不能进行关系运算

【答案】C

【解析】B 选项中,字符常量在内存中占 1 个字节;A 选项中,转义字符是以反斜线\开头,后跟一个或多个字符;D 选项中,字符常量可以进行关系运算,不能参与数值运算。因此 C 选项正确。

10. 以下叙述中正确的是()。

A. 字符变量在定义时不能赋初值

B. 字符常量可以参与任何整数运算

C. 同一个英文字母的大写和小写形式代表的是同一个字符常量

D. 转义字符以@符号开头

【答案】B

【解析】A 选项中,字符变量在定义时可以赋初值;C 选项中,同一个英文字符的大写和小写形式代表的是不同的字符常量;D 选项中,转义字符是以反斜线\开头的。因此 B 选项正确。

11. C 语言中的 double 类型数据占()个字节。

- A. 4 B. 8 C. 12 D. 16

【答案】B

【解析】double 型数据在内存中占 8 个字节,float 型数据占 4 个字节,int 型数据占 2 个字节,char 型数据占 1 个字节,所以选 B。

12. 若有定义语句“char c='\101';”,则变量 c 在内存中占()个字节。

- A. 1 B. 2 C. 3 D. 4

【答案】A

【解析】本题考查字符变量的存储空间,每个 char 型变量被分配 1 个字节的内存空间,故选 A。

13. 若有以下程序:

```
#include <stdio.h>
main()
{char c1,c2;
c1='C'+ '8'-'3'; c2='9'-'0';
printf(" %c %d\n",c1,c2);
}
```

则程序的输出结果是()。

- A. H '9' B. H 9 C. F '9' D. 表达式不合法,输出无定值

【答案】B

【解析】本题考查字符变量及 printf()函数的相关知识,字符变量 c1 被赋值为 'C'+ '8'-'3',即 ASCII 码的运算,67+54-49=72,即 H;字符变量 c2 被赋值为 '9'-'0',但输出时,需要注意的是,c1 以字符变量输出,而 c2 是以十进制整型变量输出。故选 B。

14. 有以下程序:

```
#include <stdio.h>
main()
{ int x=35,B; char z='B';
B=((x)&&(z<'b'));
printf(" %d\n",B);
}
```

程序运行后的输出结果是()。

A. 1

B. 0

C. 35

D. 66

【答案】A

【解析】本题考查逻辑运算符和关系运算符的相关知识,已知变量 x 为整型变量,并赋值为 35,变量 z 为字符型变量,并赋值为 'B'。语句“ $B=((x)\&\&(z<'b'))$;”中, (x) 的值为 1, 'B' 的 ASCII 码小于 'b' 的 ASCII 码,所以 $(z<'b')$ 的值也为 1, $1\&\&1$ 结果为 1, 故选 A。

15. 有以下程序:

```
#include <stdio.h>
main()
{   int sum,pad,pAd;
    sum=pad=5;
    pAd=++sum,pAd++,++pad;
    printf(" %d\n",pad);
}
```

则程序的输出结果是()。

A. 5

B. 6

C. 7

D. 8

【答案】B

【解析】C 语言区分大小写,首先 pad 与 pAd 是两个不同的变量。将 5 赋值给两个变量,根据赋值运算符的优先级高于逗号运算符,所以计算 $pAd=++sum$,此时 pAd 为 6, sum 也为 6,然后计算 $pAd++$, pAd 的值变为 7, $++pad$ 后 pad 的值变为 6,所以选择 B。

16. 有以下程序:

```
#include <stdio.h>
main()
{   int a=0,b=0,c=0;
    c=(a-=++a),(a+=b,b+=4);
    printf(" %d, %d, %d\n",a,b,c);
}
```

则程序运行的输出结果是()。

A. 0,4,0

B. 0,4,4

C. 1,4,1

D. 1,4,4

【答案】A

【解析】逗号运算符的结合性从左到右,因此逗号表达式将从左到右计算,且逗号运算符的优先级最低,所以先计算 $c=(a-=++a)$ 部分。其中,赋值运算符从右往左计算,先执行 $++a$, a 自增 1 后再赋值,所以 a 的值为 1,执行 $a=a-1$,即 $a=1-1$, a 的值为 0,并将 0 赋给变量 c 。第二部分 $(a+=b,b+=4)$,先执行 $a+=b$, $a=0+0$,即 a 的值为 0,然后执行

$b+=4, b=0+4$, 即 b 的值为 4, 因此输出结果为 A。

17. 有以下程序:

```
#include <stdio.h>
main()
{int a=0,b=0,c=0,d;
c=(a+=b,,b+=a); /* 第 4 行 */
d=c; /* 第 5 行 */
; /* 第 6 行 */
;printf("%d,%d,%d\n",a,b,c); /* 第 7 行 */
}
```

编译时出现错误,则出错的行是()。

A. 第 4 行 B. 第 5 行 C. 第 6 行 D. 第 7 行

【答案】A

【解析】第 4 行逗号表达式中间的第二个表达式为空,是不合法的,可以去掉写成“ $a+=b, b+=a;$ ”,也可以在里面补充一个表达式,所以选择 A。

18. 有以下程序:

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

则程序的输出结果是()。

A. 0 B. 9 C. 3 D. -12

【答案】D

【解析】赋值运算符的结合性是从右到左的,优先级低于 $+$ 、 $-$ 、 $*$ 。首先计算 $a*a$, 结果为 9, 执行 $a=a-9$, a 的值为 -6, 然后执行 $a+=a$, 即 $a=a+a=-6+(-6)$, 所以结果为 -12。

19. 有以下程序:

```
#include <stdio.h>
main()
{int b=10,a=-11;
a%=b%=4;
printf("%d %d\n",a,b);
}
```

则程序的输出结果是()。

- A. 1 2 B. -1 2 C. -1 -2 D. 1 -2

【答案】B

【解析】先计算 $b\%4=4$, 即 $b=b\%4$, 结果为 2, 然后计算 $a\%2=2$, 结果为 -1, 所以输出为 -1 2, 故选 B。

20. 以下叙述中正确的是()。

A. a 是实型变量, $a=10$ 在 C 语言中是允许的, 因此可以说, 实型变量中可以存放整型数

B. 在赋值表达式中, 赋值号的右边可以是变量, 也可以是任意表达式

C. 若有 “int a=4, b=9;”, 执行了 $a=b$ 后, a 的值变为 b 的值, b 的值变为 0

D. 若有 “int a=4, b=9;”, 执行了 “ $a=b; b=a;$ ” 后, a 的值为 9, b 的值为 4

【答案】B

【解析】A 选项中的 a 是实型变量, $a=10$ 在 C 语言中是允许的, 但是实型变量中不可以存放整型数。C 选项中执行 $a=b$ 后, a 的值变为 b 的值, 为 9, b 的值不变, 仍为 9。D 选项中执行 $a=b$ 后, a 的值为 9, b 的值为 9, 执行 $b=a$ 后, a 的值为 9, b 的值也为 9, 所以选择 B。

结构化程序设计



工作任务

- 深入理解 C 语言的三种结构。
- 掌握 C 语言程序选择分支结构的编写和调试方法。
- 掌握 C 语言程序循环结构的编写和调试方法。



项目要点

- 掌握 C 语言选择分支语句的用法。
- 掌握 C 语言循环语句的用法。

任务 1 编写面积计算器

小刘同学学习了 C 语言后,打算编写一个程序完成一些数学公式的计算,如计算长方形面积、三角形面积、圆的面积等。

小刘同学想,我可以先运行一个欢迎界面,在欢迎界面里选择需要计算面积的图形形状,然后根据提示输入计算所需的参数。

欢迎界面如下:

```
-----
* *                欢迎使用我的面积计算器                * *
* *                请选择计算图形的形状                    * *
* *                1: 矩形                                    * *
```

* *

2: 三角形

* *

* *

3: 圆形

* *

* *

4: 梯形

* *

1. 编程思路

程序的功能主要包括显示欢迎界面,根据输入的数字判断计算的图形再进行计算。欢迎界面使用多行打印,使用 scanf 语句获得输入的数字;根据数字值进入 select 分支语句进行面积的求解。

2. 程序代码

小刘同学编写的程序代码如下:

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
void main()
{ //打印第一行
    for(int i=0;i<=50;i++)
        printf("-");
    printf("\n");
    printf(" * * \t\t 欢迎使用我的面积计算器\t\t * * \n");
    printf(" * * \t\t 请选择计算图形的形状\t\t * * \n");
    printf(" * * \t\t\t t1: 矩形\t\t\t * * \n");
    printf(" * * \t\t\t t2: 三角形\t\t * * \n");
    printf(" * * \t\t\t t3: 圆形\t\t\t * * \n");
    printf(" * * \t\t\t t4: 梯形\t\t\t * * \n");
    for(int i=0;i<=50;i++)
        printf("-");
    int sharp=0;
    scanf("%d",&sharp);
    system("cls");
    switch(sharp){
    case 1:
        float a;
        float b;
```

```

scanf("请输入矩形的长和宽: %f, %f\n", &a, &b);
printf("矩形的面积= %f\n", a * b);
break;
case 2:
float c;
float d;
float e;
float s;
float area;
scanf("请输入三角形的三边: %f, %f, %f\n", &c, &d, &e);
s = (float)0.5 * (c + d + e);
area = (float)sqrt(s * (s - a) * (s - b) * (s - c));
printf("矩形的面积= %f\n", area);
break;
case 3:
float r;
scanf("请输入圆的半径: %f\n", &r);
printf("圆的面积是: %f\n", 3.1415 * r);
break;
case 4:
float f;
float g;
float h;
scanf("请输入梯形的上底和下底: %f, %f\n", &f, &g);
scanf("请输入梯形的高: %f\n", &h);
printf("梯形的面积是: %f\n", (f + g) * h * 0.5);
break;
default : break;
}
}

```

3. 运行结果

小刘同学的程序运行结果如图 3-1~图 3-5 所示。

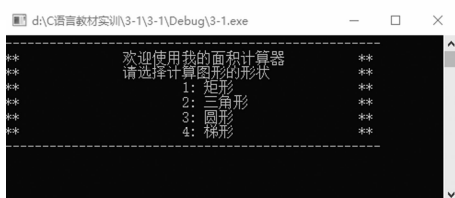


图 3-1 我的计算器欢迎界面

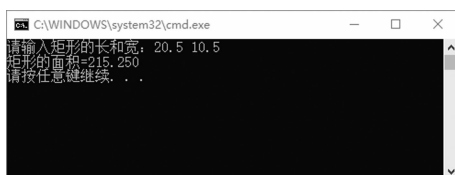


图 3-2 计算矩形的面积

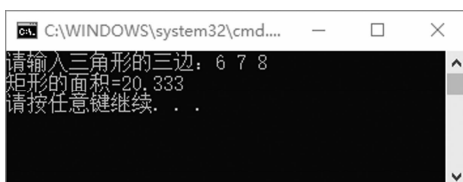


图 3-3 计算三角形的面积

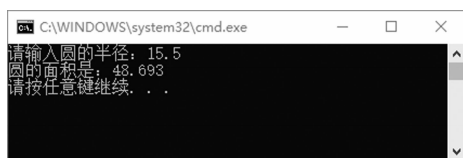


图 3-4 计算圆的面积



图 3-5 计算梯形的面积

4. 改进与拓展

你能将小刘同学的程序在你的计算机中运行出来吗? 计算结果还有许多需要优化的问题,如小数位过多,请你试着把这些问题进行优化。

另外,在每次运算之后程序会自动退出,是否能够改进程序,使得按某一键返回欢迎界面呢。大家在数学学习过程中,一定遇到过很多同类的数学公式,能否也编写一个类似的程序来帮助我们完成计算。

任务2 用枚举算法求解百元百鸡问题

中国古代数学家著有古典数学问题《算经》，其中，著名的“百钱百鸡”问题的叙述如下：“鸡翁一，钱值五；鸡母一，钱值三；鸡雏三，钱值一；百钱买百鸡，问翁、母、雏各几何？”这个问题翻译过来就是：“一只公鸡值五元钱，一只母鸡值三元钱，三只小鸡值一元钱；请问用一百元钱买一百只鸡，公鸡、母鸡和小鸡各有多少只？”

1. 编程思路

假设公鸡、母鸡和小鸡的个数分别是 x 、 y 、 z ，那么买公鸡的钱数为 $5x$ ，买母鸡的钱数为 $3y$ ，买小鸡的钱数为 $z/3$ ；再由题意 x 、 y 、 z 的和为 100，因此，可以得到该问题的数学模型为

$$\begin{aligned} 5x + 3y + \frac{z}{3} &= 100 \\ x + y + z &= 100 \end{aligned}$$

这是有一个补丁方程组，因为鸡的数量一定是整数，因此可以在取值范围内逐一变化 x 、 y 、 z 的值，可以让计算机穷举所有的可能情况，从而找到所有的解。

在本题中，如果 100 元钱全部买成公鸡，最多买 20 只，因此 x 的取值范围为 $0 \sim 20$ ；同理， y 的取值范围为 $0 \sim 33$ 。

具体求解过程如下：

- (1) 公鸡的数 x 取 0。
- (2) 母鸡的数 y 取 0。
- (3) 通过关系式 $z = 100 - x - y$ ，计算小鸡的数 z 。
- (4) 判断 $15x + 9y + z$ 的值是否等于 300，若相等，则输出此时公鸡、母鸡和小鸡的个数。
- (5) y 增 1，若 $y \leq 33$ ，转到(2)，否则转到(6)。
- (6) x 增 1，若 $x \leq 20$ ，转到(1)，否则转到(7)。
- (7) 结束求解过程。

由此可见，求解的过程就是当 x 从 0 逐个变化到 20 时，重复处理①、②、③、④，这一过程可表示成如下形式：

```
for(x=0;x<=20;x++)
```

而处理①、②、③、④的代码是：

```
for(y=0;y<=33;y++)
{
    z=100-x-y;
    if(15*x+9*y+z==300)
        printf("cock= %-3d hen= %-3d chicken= %-3d\n",x,y,z);
}
```

2. 程序代码

```
#include <stdio.h>

void main()
{
    int x=0,y=0,z=0;
    printf("Possible solutions to buy 100 fowls with 100 yuan is:\n");
    for(x=0;x<=20;x++)
        for(y=0;y<=33;y++)
        {
            z=100-x-y;
            if(15*x+9*y+z==300)
                printf("cock= %-3d hen= %-3d chicken= %-3d\n",x,y,z);
        }
}
```

运行程序,得到图 3-6 所示的结果。

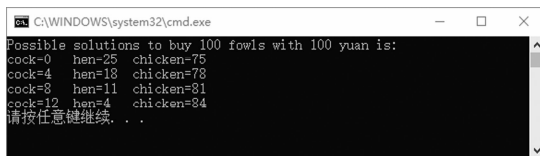


图 3-6 百元百鸡问题求解程序的运行结果

3. 归纳总结

百元百鸡问题求解程序使用一个循环内又包含另外一个完整循环的结构,这种循环称为双重循环,这种结构在 C 语言中非常常见。因为计算机的优势之一就是完成重复计算,利用循环体可以解决很多数学中的未解难题。

在书写多重嵌套结构时,一定要注意程序的缩进。只有这样才能清晰地看出程序的结构,增强程序的可读性。



应试宝典

【考点解析】

C 语言的语句基本形式主要包括顺序结构、选择分支结构和循环结构。

1. 格式输出函数 printf() 和格式输入函数 scanf()

(1) printf() 函数是 C 语言提供的标准输出函数,它的作用是向终端按指定格式输出若干个数据。

printf() 函数的一般形式为:

printf(格式控制字符串,输出表列);

①格式控制字符串。用双引号引起来的字符串是格式控制字符串,它包括以下两种信息:

- 格式转换说明,由“%”和格式说明符构成。
- 需要原样输出的字符也写在格式控制字符串内。

②格式字符。可在“%”与格式字符之间插入“宽度说明”、左对齐符号“-”、前导零符号“0”等。

- d 格式符,用来对十进制数进行输入输出。
- o 格式符,表示以八进制形式输出整数。
- x 格式符,表示以十六进制形式输出整数。
- u 格式符,用来输出 unsigned 型数据,即输出无符号的十进制数。
- c 格式符,用来输出一个字符。
- s 格式符,用来输出一个字符串。
- f 格式符,用来输出实数(包括单、双精度),以小数形式输出,使整数部分全部输出。
- e 格式符,以指数形式输出实数。
- g 格式符,用来输出实数。

③使用 printf() 函数时的注意事项。

- 在格式控制字符串中,格式说明与输出项从左到右在类型上必须一一对应匹配。
- 在格式控制字符串中,格式说明与输出项个数要相等。
- 在格式控制字符串中,可以包含任意的合法字符(包括转义字符),这些字符在输出时被原样输出。

- 如果要输出“%”,则应该在格式控制字符串中用两个连续的百分号“%%”来表示。

(2)scanf()函数的一般形式是:

scanf(格式控制字符串,地址表列);

其中,scanf 是函数名,“格式控制字符串”的含义同 printf() 函数,“地址表列”由若干个变量地址组成,既可以是变量的地址,也可以是字符串的首地址。

scanf()函数中的格式说明也是以%开始,以一个格式字符结束,中间可以加入附加的字符。

- 对 unsigned 型变量的数据,可以用%d、%o、%x 格式输入。
- 在 scanf() 函数中,格式字符前可以用一个整数指定输入数据所占宽度,但对于输入实型数,则不能指定其小数位的宽度。
- 在格式控制串中,格式说明的个数应该与输入项的个数相等,且要类型匹配。

使用 scanf() 函数时的注意事项:

- scanf() 函数中的输入项只能是地址表达式,而不能是变量名或其他内容。
- 在用“%c”格式输入字符时,空格字符和转义字符都可作为有效字符输入。
- 在输入数据时,若实际输入数据少于输入项个数,scanf() 函数会等待输入,直到满足

条件或遇到非法字符才结束;若实际输入数据多于输入项个数,多余的数据将留在缓冲区备用,作为下一次输入操作的数据。

- 在输入数据时,遇到以下情况时认为输入结束:空格、“回车”或“跳格”(“Tab”键),上述字符可统一称为间隔符。

2. 选择分支结构

(1)关系运算符和关系表达式。

①关系运算符及其优先级。C 语言提供了 6 种关系运算符:小于(<)、小于等于(<=)、大于等于(>=)、大于(>)、等于(==)、不等于(!=)。这些运算符中,前四种关系运算符(<、<=、>=、>)的优先级相同,后两种关系运算符(==、!=)的优先级相同。前面的四种高于后两种。关系运算符的优先级低于算术运算符,高于赋值运算符。

②关系表达式。由关系运算符连接成的表达式称为关系表达式。关系运算符的两边可以是任意合法的表达式。关系运算符的结果是一个整数值 0 或 1,用非零来表示“真”,用零值来表示“假”。当关系运算符两边的类型不一致时,系统将自动进行转化。

(2)逻辑运算符和逻辑表达式。C 语言提供了三种逻辑运算符:逻辑与(&&)、逻辑或(||)和逻辑非(!)。其中,“&&”和“||”是双目运算符,而“!”是单目运算符,要求必须出现在运算对象的左边。

“!”运算符的优先级最高,整个优先级的排序是:“!”>算术运算符>关系运算符>“&&”>“||”>赋值运算符。需要注意的是,“&&”运算和“||”运算的优先级是不同的。

逻辑表达式由逻辑运算符和运算对象组成。参与逻辑运算的对象可以是一个具体的值,还可以是 C 语言中任意合法的表达式。

逻辑表达式的运算结果为 1(真)或 0(假)。

- A&&B 运算中,只有 A、B 同时为真时结果才为真。
- A||B 运算中,只有 A、B 同时为假时结果才为假。

关系运算符不能连用,即不能使用 $0 < x < 10$,但可以写成 $0 < x \&\& x < 10$ 。

(3)if 语句。if 语句包括以下几种形式:

①if(表达式) 语句。

if 是 C 语言的关键字,另外表达式两侧的括号不可少,且只能是圆括号。紧跟着的语句称为 if 子句,如果在 if 子句中需要多个语句,则应该使用大括号{}把一组语句括起来构成复合语句。

②if(表达式)语句 1。

else 语句 2

③if(表达式)语句 1。

else if(表达式 2) 语句 2

else if(表达式 3) 语句 3

else if(表达式 4) 语句 4

.....

else if(表达式 m) 语句 m

else 表达式 n

语句 1 是 if 的子句,语句 2……语句 m 是 else 的子句,这些子句在语法上要求是一条语句,当需要执行多条语句时,应该使用大括号{}把这些语句括起来组成复合语句。else 必须与 if 配对,共同组成 if……else 语句。

在 if 语句中又包含一个或多个 if 语句结构,称为 if 语句的嵌套。

④条件运算符。简单的选择结构还可以使用条件运算符(?:)构成。条件运算符的一般形式为:

表达式 1? 表达式 2:表达式 3

条件运算符的计算过程是先求表达式 1 的值,当表达式 1 的值是非 0 时,以表达式 2 的值作为整个条件表达式的值;当表达式 1 的值是 0 时,以表达式 3 的值作为整个条件表达式的值。

条件运算符的优先级高于赋值运算符,但低于逻辑运算符、关系运算符和算术运算符。

⑤switch 语句。switch 语句是 C 语言提供的多分支选择语句,用来实现多分支选择结构。其一般形式如下:

switch(表达式)

{

case 常量表达式 1:语句 1

case 常量表达式 2:语句 2

.....

case 常量表达式 n:语句 n

default:语句 n+1

}

在 switch 结构中,switch 后面括号内的表达式可以是 C 语言中任意合法的表达式。case 与其后面的常量表达式的类型必须与 switch 后面表达式的类型相匹配,且各 case 语句标号的值各不相同,不能重复。当 switch 后面的表达式与 case 后面的常量相等时,程序从这个 case 开始执行直到完成整个结构或遇到 break 语句为止。

default 是 C 语言的关键字,起到标号的作用。代表除了以上所有 case 标号之外的那些标号,default 标号可以出现在语句体中任何标号位置上,也可以没有。

case 语句标号后的语句 1、语句 2 等,可以是一条语句,也可以是若干条语句,在必要时,case 语句标号后的语句也可省略不写。

3. 循环结构

(1)while 循环语句。while 循环语句的一般形式为:

while(表达式) 循环体

其中,while 是 C 语言关键字,后面的表达式是循环条件,用来控制循环体是否执行。循环体

只能是一条可执行语句,当多项操作需要多次重复操作时,可以使用复合语句。

while 循环语句的执行过程是:

第一步,计算紧跟 while 后括号中表达式的值,当表达式的值为非 0 时,接着执行 while 语句中的内嵌语句;当表达式的值为 0 时,跳过该 while 语句,执行该 while 结构后的其他语句。

第二步,执行循环体内嵌语句。

第三步,返回执行第一步,直到条件不满足,即表达式的值为 0 时,退出循环,while 结构结束。

while 循环语句的特点是先对表达式进行条件判断,后执行语句,即先判断后执行。

(2)do...while 语句。do...while 语句的一般形式为:

```
do{  
    循环体语句  
}while(表达式);
```

在 do...while 语句中,do 是 C 语言关键字,必须和 while 联合使用,不能独立出现。do...while 循环由 do 开始,用 while 结束。while 后面的圆括号中的表达式用来控制循环是否执行,且括号不能省略。在 do 和 while 之间只能是一条语句,当需要执行多条语句时,可以使用复合语句。

do...while 语句的执行过程是:先执行一次循环体语句,执行完后,判别 while 后面的表达式的值,当表达式的值为非零(真)时,程序流程返回,去重新执行循环体语句。如此循环,直至表达式的值为零(假)为止。

do...while 语句的特点是先执行循环体一次,然后判断循环条件是否成立,即先执行后判断。

(3)for 语句。for 语句的一般形式为:

```
for(表达式 1;表达式 2;表达式 3)
```

for 语句的圆括号中通常有 3 个表达式,用于 for 循环的控制。for 语句中的表达式可以部分或全部省略,但两个“;”是不可省略的。各个表达式之间用“;”隔开,且圆括号不可省略。按照语法规则,循环体只能是一条语句,当需要完成多项操作时,可以使用复合语句。

for 语句的执行过程是:先求表达式 1 的值,再求表达式 2 的值,若其值为真,则执行 for 语句中指定的内嵌语句,然后求表达式 3 的值,若表达式 3 的值为假,则退出循环。若表达式 3 的值为真,则继续执行循环体和表达式 3,直至表达式 1 为假为止。

4. 几种循环的比较

while 和 do...while 循环,只在 while 后面指定循环条件,循环体内应包含使循环趋于结束的语句,for 中使循环趋于结束的操作可以包含在表达式 3 中。另外,由 while 完成的循环用 for 循环都能完成。

【真题全解】

1. 有以下程序:

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

程序的输出结果是()。

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

【答案】C

【解析】C语言中用“%%”打印输出字符“%”，所以%%d,输出为%d两个普通字符，而不是格式控制符“%d”的含义，所以打印结果为C。

2. 以下叙述中正确的是()。

- A. 在 scanf 函数的格式控制字符串中,必须有与输入项一一对应的格式转换说明符
 B. 只能在 printf 函数中指定输入数据占的宽度,而不能在 scanf 函数中指定输入数据占的宽度
 C. scanf 函数中的格式控制字符串是提示程序员的,输入数据时不必管它
 D. 复合语句也被称为语句块,它至少要包含两条语句

【答案】A

【解析】在 printf 和 scanf 函数中都可以指定数据的宽度,所以 B 选项是错误的。scanf() 的格式控制字符串可以使用其他非空白字符(如逗号),但在输入时必须输入这些字符,以保证匹配,所以 C 选项错误。复合语句可以由任意多条语句构成,可以一条没有,所以 D 选项错误。

3. 有以下程序:

```
#include <stdio.h>
main()
{int x=1,y=0,a=0,b=0;
switch(x)
{case 1: switch(y)
        {case 0:a++;break;
         case 1:b++;break;
         }
  case 2:a++;b++;break;
}
printf("a= %d,b= %d\n",a,b);
}
```

程序的输出结果是()。

- A. $a=1, b=0$ B. $a=1, b=1$ C. $a=2, b=1$ D. $a=2, b=2$

【答案】C

【解析】本题重点考查 switch 语句,switch 语句中 case 分支后面如果没有 break,那么就会顺序执行下面的 case,直到遇到 break 跳出。由于变量 a 和变量 b 均被赋初值为 0,当程序执行完第二层 switch 的“case 0: $a++$; break;”后,第二层 switch 结束,a 的值为 1,继续执行第一层 switch 的语句“case 2: $a++$; $b++$; break;”,a 的值为 2,b 的值为 1。因此 C 选项正确。

4. 有以下程序:

```
#include <stdio.h>
main()
{ int k,j,s;
  for(k=2;k<6;j++)
  { s=1;
    for(j=k;j<6;j++) s+=j;
  }
  printf("%d\n",s);
}
```

程序的输出结果是()。

- A. 6 B. 10 C. 24 D. 40

【答案】B

【解析】本题重点考查 for 语句循环嵌套,首先 $k=2$,进行一次循环后,k 自增两次,变成 4,这时 s 为 1,即当 $k=4$ 时, $s=1, j=4$,执行内层循环, $s=1+4, j$ 自增 1 后等于 5, $s=5+5, j$ 再自增 1 后为 6,跳出内层循环。k 自增两次后等于 6,条件不成立跳出外层循环。最后输出 s 的值,即 10。因此 B 选项正确。

5. 以下叙述中正确的是()。

- A. 在 C 语言中,逻辑真值和假值分别对应 1 和 0
 B. 关系运算符两边的运算对象可以是 C 语言中任意合法的表达式
 C. 对于浮点变量 x 和 y,表达式 $x==y$ 是非法的,会出现编译错误
 D. 分支结构是根据算术表达式的结果来判断流程走向的

【答案】B

【解析】A 选项中逻辑真值对应非 0;C 选项中,表达式 $x==y$ 是合法的;D 选项中,分支结构的流程走向是根据表达式的值,并不仅仅是算术表达式的值。因此 B 选项正确。

6. 与数学表达式 $x \geq y \geq z$ 对应的 C 语言表达式是()。

- A. $(x >= y) \parallel (y >= x)$

- B. $(x > y > z)$
 C. $(x > y) \& (y > z)$
 D. $(x > y) \& \& (y > z)$

【答案】D

【解析】本题考查关系表达式和逻辑表达式, y 大于等于 x 同时 y 小于等于 z , $x <= y$ 和 $y <= z$ 是逻辑与的关系, 因此 D 选项正确。

7. 以下叙述中正确的是()。

- A. 逻辑或运算符 $||$ 的优先级比算术运算符高
 B. C 语言的关系表达式 $0 < x < 10$ 等价于 $(0 < x) \& \& (x < 10)$
 C. 逻辑非运算符 $!$ 的优先级是最低的
 D. 由 $\& \&$ 构成的逻辑表达式与由 $||$ 构成的逻辑表达式都有短路现象

【答案】D

【解析】B 选项中, $(x > 0) \& \& (x < 10)$ 的值是 1 和 0, 条件是 $x > 0$ 且 $x < 10$ 。但是 $0 < x < 10$ 这种写法的结果任何时候都是 1, 因为 $0 < x$ 为真时值为 1, 为假时值为 0, 但无论是 1 还是 0, 都是小于 10 的; C 选项中, 逻辑“非”(运算符 $!$) 的运算级别最高; A 选项中, 算术运算符优先级较高, 关系运算符和逻辑运算符优先级较低。D 选项中所谓的短路现象是指只要能够明确无误地确定整个表达式的值, 就不再计算表达式余下部分, 即整个逻辑表达式后的部分有可能不会被运算, 因此 D 选项正确。

8. 有以下程序:

```
#include <stdio.h>
main()
{ int a=1,b=2,c=3,d=4;
  if((a=2) || (b=1)) c=2;
  if((c==3) && (d=-1)) a=5;
  printf("%d, %d, %d, %d\n", a, b, c, d);
}
```

程序的输出结果是()。

- A. 1, 2, 3, 4 B. 2, 1, 2, -1 C. 5, 1, 2, -1 D. 2, 2, 2, 4

【答案】D

【解析】第一个 if 语句的判断条件是逻辑或表达式, 所以只要有一个运算对象为真, 结果就为真, 且如果前面的已经为真, 后面的表达式不再进行判断。执行第一个 if 判断条件时 a 的值为 2, 整个表达式的值为真, b 的值不变, c 的值为 2。第二个 if 语句的判断条件为逻辑与表达式, 只有两个运算对象均非零才为真, 第一个表达式 $c == 3$ 不成立, 不再判断第二个

运算对象,所以不执行 $d=-1$ 操作和 $a=5$ 操作。所以结果为 D。

9. 有以下程序:

```
#include <stdio.h>
main()
{ int a=1,b=2,c=3,d=4,r=0;
  if(a!=1) ; else r=1;
  if(b==2) r+=2;
  else; if(c!=3) r+=3;
  else; if(d==4) r+=4;
  printf("%d\n",r);
}
```

程序的输出结果是()。

A. 10

B. 7

C. 6

D. 3

【答案】B

【解析】else 总是和与之最近的上面的 if 配对,执行第一个 if 语句,判断条件不成立,执行“else r=1;”语句。再执行第二个 if 语句中的判断条件,成立,所以执行“r+=2”,r 的值为 3,再执行第三个 if 语句中的判断条件,不成立,所以执行第 7 行 else 后面的空语句“;”。再执行第四个 if 语句中的判断条件 $d==4$,成立,所以执行“r+=4”,r 的值为 7,然后打印输出。

10. 以下叙述中正确的是()。

A. 对于逻辑表达式“ $a++||b++$ ”,设 a 的值为 1,则求解表达式的值后,b 的值会发生改变

B. 对于逻辑表达式“ $a++\&\&b++$ ”,设 a 的值为 0,则求解表达式的值后,b 的值会发生改变

C. else 不是一条独立的语句,它只是 if 语句的一部分

D. 关系运算符的结果有三种:0、1 和 -1

【答案】C

【解析】A 选项和 B 选项中,||、&& 都是先计算左边表达式的值,当左边表达式的值能确定整个表达式的值时,不再计算右边表达式的值,且 &&、|| 的优先级比 ++ 高,所以题中只需判断 a 的值即可知道逻辑表达式的真假;D 选项中,关系运算符的结果有真和假两种,因此 C 选项正确。

11. 有以下程序:

```
#include <stdio.h>
main()
```

```
{double x=2.0,y;
if(x<0.0) y=0.0;
else if((x<5.0)&&(!x)) y=1.0/(x+2.0);
else if(x<10.0) y=1.0/x;
else y=10.0;
printf(" %f\n",y);
```

程序的输出结果是()。

A. 0.000000

B. 0.250000

C. 0.500000

D. 1.000000

【答案】C

【解析】本题重点考查 if 语句,变量 x 和 y 均为 double 型,由于变量 x 被赋值为 2.0,因此执行“if(x<5.0)&&(!x)”,条件为 0,继续执行 if(x<10.0),条件为 1,执行“y=1.0/x;”。依据 printf() 函数输出类型中 f 是以小数形式输出单、双精度实数。因此 C 选项正确。

12. 有以下程序:

```
#include <stdio.h>
main()
{ char a='H';
a=(a>='A' && a<='Z')?(a+32):a;
printf(" %c\n",a);
}
```

程序的输出结果是()。

A. H

B. h

C. A

D. a

【答案】B

【解析】条件运算符构成的选择结构的一般形式为:表达式 1?表达式 2:表达式 3。其求值规则为:若表达式 1 的值为真,则以表达式 2 的值作为条件表达式的值,否则以表达式 3 的值作为整个条件表达式的值。该题目中首先给字符变量 a 赋值 'H'。语句“a=(a>='A'&& a<='Z')?(a+32):a;”即将大写字母转换为小写字母,因此 B 选项正确。

13. 有以下程序:

```
#include <stdio.h>
main()
{ int s=0,n;
for(n=0;n<4;n++)
{ switch(n)
{ default: s+=4;
```

```

        case 1: s+=1;
        case 2: s+=2;
        case 3: s+=3;
    }
}
printf("%d\n",s);
}

```

程序的输出结果是()。

- A. 6 B. 18 C. 10 D. 24

【答案】D

【解析】第一次 for 循环,n 的值为 0,所以从 default 后面的语句开始执行,s+=4,s+=1,s+=2,s+=3,s 的值为 10。进入第二次 for 循环,n 的值为 1,所以执行 s+=1,s+=2,s+=3,s 的值为 16。进入第三次 for 循环,n 的值为 2,所以执行 s+=2,s+=3,s 的值为 21。进入第四次 for 循环,n 的值为 3,所以执行 s+=3,s 的值为 24。

14. 若以下选项中的变量 a、b、y 均已正确定义并赋值,则语法正确的 switch 语句是()。

- A. switch(a * a + b * b)
 { default: break;
 case 3: y=a+b;break;
 case 2:y=a-b;break;
 }
 B. switch(a+b)
 { case1:case3: y=a+b;break;
 case0:case4:y=a-b;
 }
 C. switch(a+9)
 { case a: y=a-b;
 case b:y=a+b;
 }
 D. switch a * b
 { case 10: y=a+b;
 default: y=a-b;
 }

【答案】A

【解析】B 选项中,case1 和 case2 有错误;C 选项中,case a 和 case b 不正确,case 后面应

该跟常量表达式;D 选项中,switch a * b 有误,应该用括号把 a * b 括起来,因此 A 正确。

15. 有以下程序:

```
#include <stdio.h>
main()
{
    int a=-2,b=0;
    while(a++) ++b;
    printf("%d,%d\n",a,b);
}
```

程序的输出结果是()。

A. 1,2

B. 0,2

C. 1,3

D. 2,3

【答案】A

【解析】第一次 while 判断条件中,a++先用后加,a 为-2 不为 0,条件为真,执行 a 的自加与 ++b 操作,a 的值变为-1,b 的值变为 1。第二次 while 判断后,a 的值变为 0,b 的值变为 2。第三次 while 判断条件为假,但是要让 a 执行自加操作,即值为 1,跳出循环。打印 1 和 2。

16. 有以下程序:

```
#include <stdio.h>
main()
{ int a=-2,b=0;
  while(a++)b++;
  printf("%d,%d\n",a,b);
}
```

程序的输出结果是()。

A. 0,2

B. 0,3

C. 1,3

D. 1,2

【答案】D

【解析】本题重点考查 while 语句,变量 a 和 b 分别被赋初值为-2 和 0,while 语句的语义是计算表达式的值,当值为真(非 0)时,执行循环体语句。a++即 a 参与运算后,a 的值再自增 1。++b 即 b 自增 1 后再参与运算。当表达式的值为 0 时,退出循环,最后输出 a 和 b 的值为 1 和 2,故选 D。

17. 有以下程序:

```
#include <stdio.h>
main()
```

```

{int a=6,b=0,c=0;
for(;a&&(b==0);)
    { b+=a; a-=c++;}
printf("%d,%d,%d\n",a,b,c);
}

```

程序的输出结果是()。

A. 5,6,0

B. 6,0,0

C. 6,6,1

D. 5,6,1

【答案】C

【解析】本题重点考查 C 语言中的 for 语句和逻辑表达式。本题中的 for 语句省略了表达式 1(循环变量赋初值)和表达式 3(循环变量增量)。for(;a&&(b==0);){ b+=a; a-=c++;}相当于 while(a&&(b==0)){ b+=a; a-=c++;}。因此 C 选项是正确的。

18. 有以下程序:

```

#include <stdio.h>
main()
{ int a,b;
  for(a=1,b=1;a<=100;a++)
  { if(b>=20) break;
    if(b%3==1){b=b+3;continue;}
    b=b-5;
  }
  printf("%d\n",a);
}

```

程序的输出结果是()。

A. 10

B. 9

C. 8

D. 7

【答案】C

【解析】首先注意 for 循环的控制条件,当 $b \geq 20$ 或 $a > 100$ 时,则跳出 for 循环,即 $b \leq 20$ 或 $a \leq 100$ 时执行 for 循环;第一次进入 for 循环, $a=1, b=1$ 均满足循环条件,但 $b \% 3 == 1$ 条件满足,故执行 $b=b+3$,得到 4,注意有 continue,所以后面的语句不执行,直接跳到 $a++$ 这个语句,所以第一次循环完后 $a=2, b=4$;进入第二轮循环, $b \% 3 == 1$ 也是满足的,故再次执行 $b=b+3$,此轮循环执行之后 $a=3, b=7$,进入下一轮。此后与前面循环同理,都是 $b \% 3 == 1$ 满足,因为每次都是加 3,而之后又模除 3,且都跳过 for 后面的语句直接执行 $a++$,所以,一直循环到 $b=22$ 跳出 for 循环。此时 a 为 8。综合起来就是,每次循环 b 增加 3, a 增加 1,且当 $b > 22$ 时跳出循环,结束程序。所以 b 从 1 增加到 22,有 $(22-1)/3=7$,所以 $a=1+7=8$,因此 C 选项正确。

19. 有以下程序:

```
#include <stdio.h>
main()
{int i,sum;
  for (i=1;i<6;i++)sum+=i;
  printf("%d\n",sum);
}
```

程序的输出结果是()。

A. 随机值 B. 0 C. 15 D. 16

【答案】A

【解析】本题重点考查 for 语句,变量没有初始化的情况下,默认值与编译器有关,不同的编译器有不同的值。因此 A 选项正确。

20. 有以下程序:

```
#include <stdio.h>
main()
{int a=-2,b=0;
  do{++b;} while(a++);
  printf("%d,%d\n",a,b);
}
```

程序的输出结果是()。

A. 1,3 B. 0,2 C. 1,2 D. 2,3

【答案】A

【解析】本题重点考查 do...while 语句及 do...while 的循环语句保证会执行一次,++b 即变量 b 自增 1 后再参与运算,而 a++ 即变量 a 参与运算后再自增 1。所以第一次循环结束后,b=1,a=-1。第二次循环后,b=2,a=0,这里先对 while 的判断为真,然后 a 自增至 0 时,还会进行第三次循环。第三次循环后,b=3,a=1。因此 A 选项正确。