

第一篇

C 语言与程序设计习题解析

第 1 章 C 语言与程序设计简介习题解析

1. 下面叙述中错误的是_____。
- A. 操作系统是裸机上的第一层软件
 - B. 操作系统是一种应用软件
 - C. 操作系统是硬件与其他软件的接口
 - D. 操作系统提供了人与计算机交互的界面

【解析】

操作系统是最基本的系统软件，是对硬件机器的首次扩充。因此，选项 B “操作系统是一种应用软件”的叙述是错误的，故选 B。

2. 下面叙述中错误的是_____。
- A. 程序设计是指设计、编制和调试程序的过程
 - B. 程序设计语言的基本功能就是描述数据和对数据进行处理
 - C. 程序是由人编写的指挥和控制计算机完成某个任务的指令序列
 - D. 程序设计语言就是高级语言，用它编写的程序可以直接在计算机上运行

【解析】

机器语言是计算机唯一能够直接识别并接受的语言。高级语言编写的程序必须先翻译（编译）成机器语言程序后才可以在计算机上执行。因此选项 D 的叙述是错误的，故选 D。

3. 下面叙述中正确的是_____。
- A. 编译程序是将高级语言程序翻译成等价的机器语言程序的程序
 - B. 机器语言因使用过于困难，故现在计算机根本不使用机器语言
 - C. 汇编语言是计算机唯一能够直接识别并接受的语言
 - D. 高级语言接近人们的自然语言，但其依赖具体机器的特性是无法改变的

【解析】

计算机能够直接识别的只有机器语言，所以选项 B “计算机根本不使用机器语言”是错误的。汇编语言也必须翻译成机器语言后才能为计算机识别并接受，因此选项 C 也是错误的。高级语言已经从具体计算机中抽象出来，摆脱了依赖具体计算机的问题，因此选项 D 也是错误的。所以只有选项 A 正确，故选 A。

4. 一个 C 语言程序由_____。
- A. 一个主程序和若干个子程序组成
 - B. 若干个函数组成
 - C. 若干个过程组成
 - D. 若干个子程序组成

【解析】

C 语言程序是由函数构成的。一个 C 语言程序至少有一个 main 函数，或者由一个 main 函数和若干个其他函数组成。故选 B。

5. 一个 C 语言程序的执行是_____。
- A. 从第一个函数开始, 到最后一个函数结束
 - B. 从第一个语句开始, 到最后一个语句结束
 - C. 从 main 函数开始, 到最后一个函数结束
 - D. 从 main 函数开始, 到 main 函数结束

【解析】

C 语言程序的执行总是从 main 函数开始, 并且执行到 main 函数结束。中间可以调用其他函数, 但最终必须在 main 函数结束处停止执行。故选 D。

6. 任何 C 语言的语句必须以_____结束。
- A. 句号“.”
 - B. 分号“;”
 - C. 冒号“:”
 - D. 感叹号“!”

【解析】

C 语言规定, 任何 C 语言的语句必须以“;”结束。故选 B。

7. C 语言程序的注释_____。
- A. 以“/*”开头且以“*/”结束
 - B. 以“/*”开头且以“/*”结束
 - C. 以“//”开头
 - D. 以“/*”开头或“//”开头

【解析】

C 语言程序的注释以“/*”开始并以“*/”结束, 在“/*”和“*/”之间书写注释的内容, 可以跨行注释。C 语言程序的注释也可以以“//”开始, 但注释的内容不能跨行, 即只能在以“//”开头至本行末尾结束, C 项仅指定注释以“//”开头而未指定到何处结束, 因此是错误的。故选 A。

8. 下面说法中正确的是_____。
- A. 若没有参数, 则函数名后面的圆括号可以省略
 - B. C 语言程序的 main 函数必须放在程序的开头
 - C. 一个 C 语言程序可以由若干个函数组成, 但必须有一个 main 函数
 - D. C 语言程序中的注释只能放在程序的开始部分

【解析】

由第 4 题可知应选 C。函数名后的圆括号“()”不可省略, 故选项 A 错误; main 函数不必放在程序的开头, 故选项 B 错误; 而注释可以出现在程序的任何地方, 故选项 D 也错误, 只有选项 C 正确。故选 C。

9. C 语言源程序名的后缀是_____。
- A. .exe
 - B. .c
 - C. .obj
 - D. .cp

【解析】

C 语言源程序名的后缀可以是“.c”或“.cpp”, 故选 B。

10. 下面叙述中错误的是_____。
- A. C 语言源程序经编译后生成后缀为.obj 的目标文件
 - B. C 程序经过编译、链接后才能形成一个真正可执行的二进制机器指令文件
 - C. 用 C 语言编写的程序称为源程序, 并以 ASCII 代码形式存放在一个文本文件中
 - D. C 语言中的每条可执行语句和非执行语句最终都将被转换成二进制机器指令

【解析】

C 语言程序中只有可执行语句会被转换成二进制机器指令，非执行语句（如定义变量类型的说明语句）不会被转换成二进制的机器指令。故选 D。

11. 下面 C 语言程序的写法是否正确？若有错误，请改正。

```
(1) #include<stdio.h>
    main()
    {
        printf("C program.\n")
    }
(2) void main
    {
        printf(C program.\n);
    }
```

【解析】

(1) 中的错误包括：①主函数名 main() 前缺少了函数类型说明；②printf 语句缺少语句结束的标志“;”。因此正确的程序应为：

```
#include<stdio.h>
void main( )
{
    printf("C program.\n");
}
```

(2) 由于一个 C 语言程序至少要将程序的运行结果输出（通常输出到显示器上），也可能还要由键盘输入数据，因此 C 语言程序开头处必须用包含命令“#include”将标准输入/输出的头文件“stdio.h”包含在程序中，这样程序的运行结果才能正常显示出来。此外，给出的主函数 main 其后应有“()”，否则是错误的。另外，printf 语句中的输出内容要用“”括起来。故正确的程序见 (1)。

12. 编写一个 C 语言程序，用于输出显示 How are you? 。

【参考程序】

```
#include<stdio.h>
void main( )
{
    printf("How are you?\n");
}
```

第 2 章 C 语言与程序设计基础习题解析

1. 下面给出的标识符中，能作为变量的标识符的是_____。

- A. for B. int C. word D. sizeof

【解析】

for、int 和 sizeof 均为 C 语言的关键字，只有 word 能作为变量的标识符。故选 C。

2. 在 C 语言中，下列属于构造类型的是_____。

- A. 整型 B. 字符型 C. 实型 D. 数组类型

【解析】

整型、字符型和实型均为 C 语言的基本类型，只有数组类型属于构造类型。故选 D。

3. 下面四个选项中，均是合法整型常量的选项是_____。

- | | | | |
|---------|-----------|---------|-----------|
| A. 160 | B. -0xcdf | C. -01 | D. -0x48a |
| -0xffff | 01a | 986.012 | 2e5 |
| 011 | 0xe | 0667 | 0x |

【解析】

C 语言的整型变量有十进制、八进制和十六进制三种表示形式。十进制整型常量与数学上的整数表示相同，没有前缀，直接由 0~9 的数字组成；八进制整型常量以数字 0 为前缀，后跟 0~7 的数字；十六进制整型常量以 0x（或 0X）为前缀，后跟 0~9、A~F（或 a~f）的数字。因此，选项 A 中的三个数均为整型常量；选项 B 中的“01a”既不是八进制整型常量又不是十六进制整型常量，因此写法是错误的；选项 C 中的“986.012”是实型常量；选项 D 中的“2e5”和“0x”，前者是实型常量，后者写法错误（0x 后无数字）。故选 A。

4. 下面四个选项中，均是合法实型常量的选项是_____。

- | | | | |
|----------|---------|---------|--------|
| A. +1e+1 | B. -.60 | C. 123e | D. -e3 |
| 5e-9.4 | 12e-4 | 1.2e-4 | 0.8e-4 |
| 03e2 | -8e5 | +2e-1 | 5.e-7 |

【解析】

C 语言实型常量的表示包括小数形式和指数形式两种：小数形式由数字和小数点“.”组成，且必须有小数点，而小数点前或后可以没有数字；指数形式的一般形式为“aEn”或“aen”；a 为十进制数（可以是小数），n 为十进制整数。因此，选项 A 中，“5e-9.4”因 e 后不能出现小数而错误；“03e2”因 e 前出现的必须是十进制数，而此处“03”为八进制数，故错误；选项 B 中的三个数均为合法的实型常量，因此是正确的；选项 C 中，“123e”因 e 后无数字而错误；选项 D 中，“-e3”因 e 前无数字而错误。故选 B。

5. 下面不合法的字符常量是_____。

- A. '018' B. '\"' C. '\\'

D. '\xcc'

【解析】

转义字符是以“\”开头的字符序列，它只代表一个字符。当“\”后跟 3 个八进制数时，表示该转义字符为 ASCII 码值等于这 3 个八进制数数值的字符。选项 A 中出现了数字 8，超出了八进制 0~7 的范围，故错误；选项 B 表示的是一个双引号“”的转义字符；选项 C 表示的是一个反斜杠“\”的转义字符；选项 D 表示一个 ASCII 码值为十六进制 cc 的字符。故选 A。

6. 在 C 语言中, 将其值可以改变的量称为变量, 变量具有的基本特征是_____。

- A. 变量名 B. 变量类型 C. 变量值 D. A~C 三项

【解析】

每个变量都有三个特征: 一是有一个变量名; 二是都有一个确定的类型; 三是变量可以存放值。故选 D。

7. 在 C 语言中, int 类型数据在内存中的存储形式是_____。

- A. ASCII 码 B. 原码 C. 反码 D. 补码

【解析】

整型数据在计算机中的存储形式是以补码来表示的, 故选 D。

8. 能够正确定义且赋值的语句是_____。

- A. int n1=n2=10; B. char c=32;
C. float f=f+1.1; D. double x=12.3E2.5;

【解析】

赋值表达式“n1=n2=10”是允许的, 但在定义变量并赋初值时, 不允许出现这种方式, 即只能逐个地定义变量并赋值, 如“int n1=10,n2=10;”, 故选项 A 错误; 选项 B 是正确的变量定义并赋值方式, 故选项 B 正确; 选项 C 在给变量 f 定义并赋值时, 所赋的值必须是常量, 而在此所赋的初值里又出现了变量 f 自身, 由于在定义前 f 的值是无法确定的, 因此无法将“f+1.1”的值赋给变量 f, 故选项 C 错误; 对于选项 D, 由于实型常量的指数形式要求“E”后必须是整数, 而在此为 2.5, 故选项 D 错误。故选 B。

9. 设有定义语句“char x1,x2,x3;”, 且给 x1、x2 和 x3 都赋字符'a', 则下面出错的一组赋值语句是_____。

- | | | | |
|------------|---------------|---------------|------------|
| A. x1='a'; | B. x1='\141'; | C. x1='\x61'; | D. x1=97; |
| x2='\x61'; | x2=0x61; | x2=97; | x2="a"; |
| x3=97; | x3=0141; | x3=0x61; | x3='\141'; |

【解析】

字符'a'的 ASCII 码值为 97, 同时又是八进制数 0141 和十六进制数 0x61, 因此, 选项 A、B 和 C 都是正确的。而选项 D 中, "a"为字符串而非字符, 因此选项 D 错误。故选 D。

10. 设有定义语句“float a=2,b=4,h=3;”, 下面表达式中与代数式 $\frac{1}{2}(a+b)h$ 计算结果不符的是_____。

- A. (a+b)*h/2 B. (1/2)*(a+b)*h C. (a+b)*h*1/2 D. h/2*(a+b)

【解析】

选项 B 中由于 1 和 2 都是整型, 即 1/2 的运算结果(舍去小数部分)为 0, 因此整个表达式的结果为 0, 即其计算结果与题目所要求的代数式计算结果不同, 因此选项 B 错误; 对于选项 C, 由于运算是由左向右进行的, 因此“(a+b)*h*1”的结果为实型, 再除以 2, 其结果仍为实型, 即不会出现两个整型数相除舍去小数部分的情况, 因此选项 C 正确; 同理, 选项 A 和选项 D 都是正确的。故选 B。

11. 设有定义语句“int a=2,b=3,c=4;”, 则下面选项中值为 0 的表达式是_____。

- A. (!a==1) && (!b==0) B. (a<b) && !c || 1

C. $a \&\& b$ D. $a \parallel (b+b) \&\& (c-a)$ **【解析】**

计算逻辑表达式要注意以下三点:

(1) 逻辑运算符优先级从高到低的顺序是“!”“&&”“||”。

(2) “&&”左边的表达式为“假”或“||”左边的表达式为“真”时就能够得出整个表达式的值,而无须再计算“&&”或“||”右边的表达式(即忽略其右边的表达式)。

(3) 逻辑值能够自动转换为整数值,即“真”=1,“假”=0,整数值也能自动转换为逻辑值,即0=“假”,非0=“真”。

选项 A 中, $(!a==1) \&\& (!b==0) \Rightarrow (!2==1) \&\& (!b==0) \Rightarrow (0==1) \&\& (!b==0) \Rightarrow 0 \&\& (!b==0) \Rightarrow 0$ (忽略“&&”右边的表达式)。

选项 B 中, $(a<b) \&\& !c \parallel 1 \Rightarrow (2<3) \&\& !4 \parallel 1 \Rightarrow 1 \&\& 0 \parallel 1 \Rightarrow 0 \parallel 1 \Rightarrow 1$ 。

选项 C 中, $a \&\& b \Rightarrow 2 \&\& 3 \Rightarrow 1$ 。

选项 D 中, $a \parallel (b+b) \&\& (c-a) \Rightarrow 2 \parallel (3+3) \&\& (4-2) \Rightarrow 1$ (忽略“||”右边的表达式)。

故选 A。

12. 当整型变量 c 的值不为 2、4、6 时, 值也为“真”的表达式是_____。

A. $(c==2) \parallel (c==4) \parallel (c==6)$ B. $(c>=2 \&\& c<=6) \parallel (c!=3) \parallel (c!=5)$ C. $(c>=2 \&\& c<=6) \&\& !(c\%2)$ D. $(c>=2 \&\& c<=6) \&\& (c\%2!=1)$ **【解析】**

选项 A 中的表达式是当 C 等于 2 或等于 4 或等于 6 时, 表达式的值为真, 否则为假, 所以不满足题意; 选项 C 中的表达式是 c 值在 2~6 之间并且不是奇数时, 表达式的值为真。这样, 满足表达式值为真的 c 只能取 2、4、6, 所以也不满足题意; 选项 D 中后面一个子表达式“(c%2!=1)”与选项 C 中的子表达式“!(c%2)”是等价的, 都是判断 c 不是奇数, 故也不满足题意; 选项 B 中, 子表达式“c>=2 && c<=6”意为 c 值取 2~6 均为真, 而子表达式“(c!=3) || (c!=5)”意为除 c=3 和 c=5 外均为真, 即选项 B 的表达式无论 c 取何值均为真。故选 B。

13. 设有定义语句“int k=0;”, 下面选项的四个表达式中与其他三个表达式的值不相同的是_____。

A. $k++$ B. $k+=1$ C. $++k$ D. $k+1$ **【解析】**

“++”运算符有前缀(前置)和后缀(后置)两种形式。选项 C 是前缀形式, 而选项 A 是后缀形式, 它们的区别是其表达式的值不同, 即前缀形式表达式的值为增 1 之后的值, 而后缀形式表达式的值为增 1 之前的值。并且, 选项 B 和选项 D 的表达式都是 k 加 1 之后的值, 故选 A。

14. 设有定义语句“int k=7; float a=2.5, b=4.7;”, 则表达式 $a+k\%3*(int)(a+b)\%2/4$ 的值是_____。

A. 2.500000

B. 2.750000

C. 3.500000

D. 0.000000

【解析】

本题的表达式计算如下：

$$a+k\%3*(int)(a+b)\%2/4=2.5+7\%3*(int)(2.5+4.7)\%2/4 \\ =2.5+1*7\%2/4=2.5+7\%2/4=2.5+1/4=2.5$$

故选 A。

15. 若有代数式 $\sqrt{n^x + e^x}$ (其中 e 仅代表自然对数的底数, 不是变量), 则下面能够正确表示该代数式的表达式的是_____。

- A. `sqrt(abs(n^x+e^x))` B. `sqrt(fabs(pow(n,x)+pow(x,e)))`
 C. `sqrt(fabs(pow(n,x)+exp(x)))` D. `sqrt(fabs(pow(x,n)+exp(x)))`

【解析】

在 C 语言中, 计算平方根应使用 `sqrt` 函数, 计算 n 的 x 次方应使用 `pow` 函数 (第一个参数是 n , 第二个参数是 x), 计算自然数 e 的 x 次方则使用 `exp` 函数, 计算绝对值应使用 `fabs` 函数。故选 C。

16. 下面关于 `scanf` 语句的叙述中, 正确的是_____。

- A. 输入项地址可以是一个实型常量, 如 `scanf("%f",3.5)`
 B. 只有格式控制字符串而没有输入项地址也能正确输入数据, 如 `scanf("a=%d,b=%d")`
 C. 当输入数据时必须指明输入项地址, 如 `scanf("%f",&f)`
 D. 由于 `scanf` 语句是给变量输入数据的, 因此输入项地址也可以是一个变量, 如 `scanf("%f",f)`

【解析】

函数 `scanf` 只能给变量输入数据, 即将数据读入到变量对应的内存单元中, 因此变量地址项必须是变量的存储地址而不能是变量名。因为 3.5 不是变量, 所以选项 A 错误; 因为没有变量地址项而无法将输入的数据读入到变量对应的内存单元中, 所以选项 B 错误; 选项 C 可以将输入的数据读入到变量 `f` 对应的内存单元中, 所以选项 C 正确; 因为变量地址项必须是变量的存储地址而不能是变量名, 所以选项 D 错误。故选 C。

17. 下面程序的功能是给 `r` 输入数据后计算半径为 `r` 的圆的面积 `s`, 但程序在编译时出错, 出错的原因是_____。

```
#include<stdio.h>
void main()
{
    int r;
    float s;
    scanf("%d",&r);
    s=π*r*r;
    printf("s=%f\n",s);
}
```

- A. 注释语句书写位置错误 B. 存放圆半径的变量 `r` 不应该定义为整型
 C. 输出语句中格式描述符非法 D. 计算圆面积的赋值语句中使用了非法变量

【解析】

在 C 语言源程序中只有注释部分和字符串常量部分可以使用任意字符。因此, 在计算

圆面积的赋值语句中出现的希腊字母“ π ”是非法字符，故选 D。

18. 以下程序执行的结果是_____。

```
#include<stdio.h>
void main()
{
    int x=102,y=012;
    printf("%2d,%2d\n",x,y);
}
```

A. 10,01 B. 02,12 C. 102,10 D. 02,10

【解析】

本题中，语句“`printf("%2d,%2d\n",x,y);`”表示输出两个十进制整数，每个整数的域宽均为 2。C 语言规定：当域宽小于数据的实际宽度时，域宽限制不起作用，而是按数据的实际宽度输出。由于 x 的值为 102 已突破原定域宽的长度 2，因此按实际宽度输出。而 y 值的八进制数 012 则应转化为十进制数输出（012 等于十进制数 10）。所以输出的结果是 102,10。故选 C。

19. 以下程序执行的结果是_____。

```
#include<stdio.h>
void main()
{
    int m=0256,n=256;
    printf("%o,%o\n",m,n);
}
```

A. 0256,0400 B. 0256,256 C. 256,400 D. 400,400

【解析】

程序中定义了两个整型变量 m 和 n，m 值用八进制数表示，n 值用十进制数表示。输出语句 `printf` 要求 m 和 n 都以八进制数输出。即 m 按原样输出为 256，n 值 256 转换为八进制数后输出为 400。注意，只有在输出的格式字符“%o”的“%”和“o”之间加上格式修饰符“#”，即输出语句变为“`printf("%#o,%#o\n",m,n);`”后输出的结果为选项 A。故选 C。

20. 以下程序执行的结果是_____。

```
#include<stdio.h>
void main()
{
    int a=666,b=888;
    printf("%d\n",a,b);
}
```

A. 错误信息 B. 666 C. 888 D. 666,888

【解析】

本题中，`printf` 语句的输出格式部分只有一个格式字符“%d”，而输出成员列表有两个成员 a 和 b，因此只能输出第一个成员 a 的值为 666。注意，程序并不因为 b 值未输出而出错。故选 B。

21. 以下程序执行的结果是_____。

```
#include<stdio.h>
void main()
{
    char a='a',b;
    printf("%c,",++a);
    printf("%c\n",b=a++);
}
```

- A. b,b B. b,c C. a,b D. a,c

【解析】

程序中的一个 `printf` 语句输出表达式“`++a`”的值，即先使 `a` 值加 1（即给字符'a'加 1 后变为字符'b'）后输出。所以第一个 `printf` 语句输出的值为 `b`。第二个 `printf` 语句输出 `b` 的值，而 `b` 为表达式“`a++`”的值，由于“`a++`”是先使用 `a` 值然后再给 `a` 加 1，即先将 `a` 的当前值'b'赋给了变量 `b` 并输出其值，然后变量 `a` 再加 1，即 `a` 值变为'c'。所以第二个 `printf` 语句输出的值为 `b`。故选 A。

22. 以下程序执行的结果是_____。

```
#include<stdio.h>
void main()
{
    int a=0,b=0;
    a=10; /*给 a 赋值
    b=20; 给 b 赋值*/
    printf("a+b=%d\n",a+b); /*输出计算结果*/
}
```

- A. `a+b=10` B. `a+b=30` C. 30 D. 出错

【解析】

在 C 语言程序中，经常用“`/*...*/`”来给程序进行注释，用来帮助理解程序，它对程序的运行没有任何作用。注意，本题程序从第 5 行右边的“`/*`”开始直到第六行末尾的“`*/`”都属于注释部分。即“`b=20;`”也是注释中的内容而并不是一条语句。因此，`b` 在执行 `printf` 语句时其值仍为 0。所以最终程序输出的结果（即 `a+b` 的值）为 `a+b=10`，故选 A。

23. 设有定义语句“`int i=2;`”，则表达式“`(i++)+(++i)+(++i)`”的值是_____。

- A. 9 B. 10 C. 11 D. 12

【解析】

本题的重点是理解在 VC++ 6.0 环境下计算机中执行加法运算时一次只能对两个数进行加法运算，即“两两相加”。在本题中，执行第一个“`i++`”，`i` 值没有发生变化，仍为 2（该操作要等到整个表达式的计算结束后才执行）；执行第二个“`++i`”，`i` 值由 2 变为 3，此时完成前两个 `i` 值的相加（注意，此时的 `i` 值为 3），即 `3+3=6`；然后，再执行第三个“`++i`”，`i` 值由 3 变为 4，即前两个 `i` 值相加的结果 6 再加上这第三个“`++i`”的值 4，最终得到的结果为 10。故选 B。

24. 试求下面语句段的输出结果。

```
int a=2;
printf("%d,",!a++&&++a);
```

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

【解析】

本题需要掌握的概念有两个:

(1) 逻辑表达式进行判断时,非 0 为真,0 为假,而运算的结果是真为 1,假为 0。

(2) 在计算逻辑表达式时,并不是所有的表达式都被求解,只有在必须执行下一个表达式才能求解整个表达式时,才继续求解下一个表达式。

在本题的逻辑与运算表达式中,只要前面有一个表达式被判定为“假”,就不再求解其后面的表达式,整个表达式的值为 0。对本题的第一个 printf 语句,由于表达式“!a++”中的 a 值非 0 (等于 2) 为真,则在逻辑非“!”的作用下变为假(其值为 0),因此逻辑与运算表达式“!a++&&++a”中的后一个表达式“++a”就不再求解,整个表达式的值为 0。所以第一个 printf 语句的输出结果为 0,并且输出后还需执行“a++”中的“++”操作,即 a 值增 1 (由 2 变为 3)。这样,第二个 printf 语句的输出结果为 3。综上所述,该程序段最终的输出结果为 0,3。

25. 有以下程序,若想从键盘上输入数据,使变量 m 的值为 123, n 的值为 456, p 的值为 789, 则正确的输入是_____。

```
#include<stdio.h>
void main()
{
    int m,n,p;
    scanf("m=%dn=%dp=%d",&m,&n,&p);
    printf("%d %d %d\n",m,n,p);
}
```

A. m=123n=456p=789✓

B. m=123 n=456 p=789✓

C. m=123,n=456,p=789✓

D. 123 456 789✓

【解析】

对于 scanf 语句,若该语句中的“格式控制字符串”中出现了格式字符之外的其他字符,则在输入数据时必须在对应的位置上照原样输入这些字符,否则会产生输入错误。由于本题使用了“scanf("m=%dn=%dp=%d",&m,&n,&p);”语句进行输入,因此根据该格式控制字符串“m=%dn=%dp=%d”应先输入“m=”,接着输入“123”,然后按格式继续输入“n=”,并在其后输入“456”,接下来继续输入“p=”,并在其后输入“789✓”。即应输入“m=123n=456p=789✓”才能实现将 123 赋给 m、456 赋给 n 及 789 赋给 p。故选 A。

26. 有以下语句段:

```
int n1=10,n2=20;
printf("_____",n1,n2);
```

要求按以下格式输出 n1 和 n2 的值,每个输出从第一列开始,请填空。

```
n1=10
n2=20
```

【解析】

在 C 语言的 printf 语句中,对于出现在“格式控制字符串”中的普通字符将按原样输出,而对出现的格式字符,在此位置上根据后面的输出列表找到对应项输出其值。本题中,

要求按以下格式输出 n1 和 n2 的值。

```
n1=10
n2=10
```

即 printf 语句的格式字符串中应首先输出字符串“n1=”，故填空处首先填入“n1=”，其后应输出 n1 的值 10，即在“n1=”后应填入格式字符“%d”（即按十进制整数输出 n1 的值），接下来应换行输出“n2=10”，故在已填入的“n1=%d”后再填入换行符“\n”。同理，应在已填入的“n1=%d\n”后继续填入“n2=%d\n”。即该空应填入 n1=%d\n n2=%d\n。

27. 计算下列表达式的值。

```
(1) (1+3)/(2+4)+8%3          (2) 2+7/2+(9/2*7)
(3) (int)(11.7+4)/4%4         (4) 2.0*(9/2*7)
```

【解析】

- (1) $(1+3)/(2+4)+8\%3=4/6+8\%3=0+8\%3=2$
 (2) $2+7/2+(9/2*7)=2+3+(4*7)=5+28=33$
 (3) $(\text{int})(11.7+4)/4\%4=(\text{int})15.7/4\%4=15/4\%4=3\%4=3$
 (4) $2.0*(9/2*7)=2.0*(4*7)=2.0*28=56.000000$

28. 阅读程序，若从键盘上输入 10 20 30↵，给出程序运行的结果。

```
#include<stdio.h>
void main()
{
    int i=0,j=0,k=0;
    scanf("%d%d%d",&i,&j,&k);
    printf("%d %d %d\n",i,j,k);
}
```

【解析】

在 scanf 语句中，格式字符“%d”表示读入一个十进制整数给对应的输入项，格式字符“%*d”表示跳过当前输入的十进制数。因此，本题输入的三个数“10 20 30↵”，与 scanf 语句的格式字符串“%d%*d%d”相对应，即先将数据 10 读入给变量 i，而由于“%*d”的作用，放弃第二个输入数据 20，将第三个数据 30 读入给变量 j。这样，由于没有数据可读，因此变量 k 仍是其定义时的初值 0。所以最后的输出结果为 10 30 0。

29. 下面程序段输出的结果是_____。

```
int i=1;
printf("%d,",i,--i,i++);
printf("%d",i);
```

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

【解析】

printf 语句对输出列表中各输出项的求值顺序是自右向左的，然后再由左到右将刚才的计算结果输出，并且若输出项个数多于格式字符的个数，则多余的输出项将不输出。因此本题第一个 printf 语句的输出项由右向左计算，即先计算 i++，由于“++”后置，因此待输出操作结束后再计算；接下来计算--i，由于“--”前置，因此先进行减 1 计算使得 i 值由 1 变到 0；最后计算 i，由于 i 不进行任何计算故其值不变仍是 0。计算结束后将由左至右输出刚才的计算结果，由于仅有一个格式符“%d”，因此只输出第一个输出项 i 的值，

即 0。注意，输出 0 后还要进行刚才第三个输出项 $i++$ 中并未计算的“++”运算，即 i 值由 0 变为 1。因此，第二个 `printf` 语句输出的 i 值为 1。即本程序段输出的结果为 0,1。

30. 阅读程序，给出程序的运行结果。

```
#include<stdio.h>
void main()
{
    int x,y,z;
    x=y=2;z=3;
    y=x++-1; printf("%d,%d\n",x,y);
    y=++x-1; printf("%d,%d\n",x,y);
    y=z--+1; printf("%d,%d\n",z,y);
    y=--z+1; printf("%d,%d\n",z,y);
}
```

【解析】

程序首先给变量 x 、 y 和 z 分别赋初值 2、2 和 3。在执行语句“ $y=x++-1;$ ”时，因为“ $x++$ ”是先参加运算然后再给 x 加 1，即 $y=2-1=1$ ，而 x 则加 1 后为 3。故 `printf` 语句输出为 3,1。执行语句“ $y=++x-1;$ ”因“ $++x$ ”是先给 x 加 1 然后再参加运算，即 $y=4-1=3$ (x 先加 1 为 4)。故 `printf` 语句输出为 4,3。同理，执行语句“ $y=z--+1;$ ”，即 $y=3+1=4$ ，而 z 减 1 后为 2。故 `printf` 语句输出为 2,4。执行语句“ $y=--z+1;$ ”，即 $y=1+1=2$ (z 值先减 1)。故 `printf` 语句输出为 1,2。

第 3 章 三种基本结构的程序设计习题解析

- 下面叙述中错误的是_____。
 - C 语言是一种结构化程序设计语言
 - 结构化程序由顺序、分支和循环三种基本结构组成
 - 使用三种基本结构构成的程序只能解决简单问题
 - 结构化程序设计提倡模块化的设计方法

【解析】

采用顺序结构、选择结构和循环结构这三种控制结构就能编写所有程序，所以选项 C 叙述错误，故选 C。

- 下面叙述中错误的是_____。
 - C 语言的语句必须以分号结束
 - 复合语句在语法上被看成一个语句
 - 空语句出现在任何位置都不会影响程序运行
 - 赋值表达式末尾加分号就构成赋值语句

【解析】

选项 A、B 和 D 的概念都是正确的。对于选项 C，空语句也是一个语句，当空语句出现在条件语句或者循环语句中时，会被当成条件语句中的子句或循环语句的循环体来看待。所以选项 C 的叙述是错误的，故选 C。

3. 在嵌套使用 if 语句时，C 语言规定 else 总是_____。

- A. 和之前与其具有相同缩进位置的 if 配对
- B. 和之前与其最近的 if 配对
- C. 和之前与其最近的且不带 else 的 if 配对
- D. 和之前的第一个 if 配对

【解析】

根据就近原则，在嵌套的 if 语句中，else 总是与它之前最近的且不带 else 的 if 配对。故选 C。

4. 设变量已正确定义，则下面能正确计算 $f=n!$ 的程序段是_____。

- | | |
|-------------------------|-------------------------|
| A. f=0; | B. f=1; |
| for(i=1;i<=n;i++) f*=i; | for(i=1;i<n;i++) f*=i; |
| C. f=1; | D. f=1; |
| for(i=n;i>1;i++) f*=i; | for(i=n;i>=2;i--) f*=i; |

【解析】

我们知道 $n! = 1 \times 2 \times 3 \times \dots \times n$ 。本题中，选项 A 给 f 赋初值 0 是错误的，因为 0 乘以任何数都为 0；选项 B 中循环变量 i 的变化范围是 $1 \sim n-1$ ，即只能求出 $(n-1)!$ ，与题意不符，因此也是错误的；选项 C 的 for 循环是一个死循环，无解；选项 D 的循环变量 i 的变化范围是 $n \sim 2$ ，虽然差了个 1，但 1 乘以任何数都不改变这个数的值，故 $n \sim 2$ 能够求出 $n!$ ，故选 D。

5. 有以下程序段：

```
int n,t=1,s=0;
scanf("%d",&n);
do
{
    s=s+t; t=t-2;
}while(t!=n);
```

为了使该程序段不陷入死循环，从键盘输入的数据应该是_____。

- A. 任意正奇数
- B. 任意负偶数
- C. 任意正偶数
- D. 任意负奇数

【解析】

do...while 语句是先执行循环体，然后判断循环条件是否为真再决定下一次的循环。本题首先给 t 赋初值为 1，然后在循环体中每次自减 2，所以判断循环条件时，t 会依次出现 -1、-3、-5、...，故只有 n 等于任意一个负奇数时才能够终止循环，否则就是死循环，故选 D。

6. 设变量 a、b、c、d 和 y 都已正确定义并赋值。若有以下 if 语句，则该语句所表示

的含义是_____。

```
if (a<b)
    if (c==d) y=0;
    else y=1;
```

- A. $y = \begin{cases} 0, & a < b \text{ 且 } c = d \\ 1, & a \geq b \end{cases}$
- B. $y = \begin{cases} 0, & a < b \text{ 且 } c = d \\ 1, & a \geq b \text{ 且 } c \neq d \end{cases}$
- C. $y = \begin{cases} 0, & a < b \text{ 且 } c = d \\ 1, & a < b \text{ 且 } c \neq d \end{cases}$
- D. $y = \begin{cases} 0, & a < b \text{ 且 } c = d \\ 1, & c \neq d \end{cases}$

【解析】

else 子句总是与前面最近的不带 else 的 if 相结合，所以本题嵌套 if 语句的含义是“在 $a < b$ 的前提下，若 c 等于 d 则执行 $y=0$ ；否则执行 $y=1$ ”。故选 C。

7. 若变量已正确定义，要求程序段完成 $5!$ 的计算，不能完成此操作的程序段是_____。

- A. for (i=1,p=1;i<=5;i++) p*=i; B. for(i=1;i<=5;i++) {p=1;p*=i;}
- C. i=1; p=1; while(i<=5){p*=i;i++;} D. i=1; p=1; do(p*=i;i++;)while(i<=5);

【解析】

选项 B 中 for 语句的循环体执行了 5 次，而每次循环时都先将 p 值重新赋 1 后再乘以当前的 i 值，因此最终的结果是 5 而不是 $5!$ ，故选 B。其余选项 A、C 和 D 的计算结果都是 $5!$ 。

8. 若有定义 float x=1.5;int a=1,b=3,c=2;，则正确的 switch 语句是_____。

- A. switch(x)
- ```
{
 case 1.0: printf("*\n");
 case 2.0: printf("**\n");
}
```
- B. switch((int)x);
- ```
{
    case 1: printf("*\n");
    case 2: printf("**\n");
}
```
- C. switch(a+b)
- ```
{
 case 1: printf("*\n");
 case 2+1: printf("**\n");
}
```
- D. switch(a+b)
- ```
{
    case 1: printf("*\n");
    case c: printf("**\n");
}
```

【解析】

switch 后面的“()”中只能是整型或字符型表达式，而选项 A 中使用的却是实型变量 x，故不正确；选项 B 在 switch 后面的“()”后多了一个分号“;”，故也不正确；switch 语句中 case 后的标号只能是常量或常量表达式而不能是变量，所以选项 D 也不正确。本题只有选项 C 是正确的，故选 C。

9. 下面叙述中正确的是_____。

- A. break 语句只能用在 switch 语句体中
- B. continue 语句的作用是使程序的执行流程跳出包含它的所有循环
- C. break 语句只能用在循环体内和 switch 语句体内
- D. 在循环体内使用 break 语句与使用 continue 语句的作用相同

【解析】

`break` 语句可以用于 `switch` 语句体内或 `while`、`do...while` 和 `for` 三种循环语句中的循环体内，作用是跳出 `switch` 语句或结束三种循环语句的执行，故选项 A 错误而选项 C 正确。`continue` 语句只能用于三种循环语句，并且其作用是结束当前的本次循环（即不执行循环体中位于 `continue` 后面的语句），而进入下一次循环，故选项 B 错误。`break` 语句和 `continue` 语句的作用不同，故选项 D 错误。故选 C。

10. 以下程序执行的结果是_____。

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

A. 1,3

B. 0,2

C. 0,3

D. 1,2

【解析】

由第 2 章的习题 24 可知：在计算逻辑表达式时，并不是所有的表达式都被求解，只有在必须执行下一个表达式才能求解整个表达式时，才继续求解下一个表达式。

本题程序中，第一次执行 `while` 循环时，条件表达式“`a++&&++b`”的“`a++`”是先判断后增 1，由于 `a` 值此时为 -2（非 0），因此必须求解表达式“`++b`”；而“`++b`”是先增 1 后判断，故 `b` 值由 0 变为 1；即条件表达式“`a++&&++b`”的结果为 1（真），因此执行一次循环体（即空语句“`;`”）后继续执行 `while` 循环。注意，在执行循环体之前还要给已进行过判断的“`a++`”加 1，这时 `a` 值由 -2 变为 -1。

第二次执行 `while` 循环时继续对表达式“`a++&&++b`”进行判断，由于 `a` 值此时为 -1（非 0），因此还必须求解表达式“`++b`”；而“`++b`”使 `b` 值由 1 变为 2，即条件表达式“`a++&&++b`”的结果为 1（真），继续执行一次循环体；同样，在执行循环体（空语句“`;`”）之前还要给已进行过判断的“`a++`”加 1，使 `a` 值由 -1 变为 0。

接下来第三次执行 `while` 循环时继续对表达式“`a++&&++b`”进行判断，由于 `a` 值此时为 0，因此无须求解表达式“`++b`”，即 `b` 值保持不变仍为 2；此时条件表达式“`a++&&++b`”的结果为 0（假），故不再执行循环体，但仍需给已进行过判断的“`a++`”加 1，使 `a` 值由 0 变为 1。故最终输出的 `a` 和 `b` 值为 1,2。

11. 阅读程序，给出程序的运行结果。

```
#include<stdio.h>
void main()
{
    int a=3,b=4,c=5,t=99;
    if(b<a&&a<c) t=a; a=c; c=t;
    if(a<c&&b<c) t=b; b=a; a=t;
    printf("%d,%d,%d\n",a,b,c);
}
```

【解析】

if 语句是当由“()”括起来的表达式为真(非 0)时,执行紧跟“()”后的第一个语句或复合语句。若表达式为假(为 0),则执行紧跟在 else 后面的第一个语句或复合语句。因为本题的 if 语句没有 else 分支,所以当表达式为真时,执行紧跟在“()”后的第一个语句,当表达式为假时,则 if 语句(包括“()”后的第一条语句)相当于一个空语句。程序中,两个 if 语句只有“t=a;”和“t=b;”是各自 if 的子句。执行中,第一个 if 语句的表达式“b<a”为假,故跳过“t=a;”而执行“a=c;c=t;”,这时各变量的值为 a=5、b=4、c=99。第二个 if 语句的表达式“a<c&& b<c”为真,所以后面的三个语句都被执行,这三个语句实际上是交换了 a 和 b 的值。所以最后输出的结果是 4,5,99。

12. 阅读程序,给出程序的运行结果。

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

【解析】

由于 C 语言中规定 else 总是与在它之前且离它最近的 if 匹配,因此程序中 else 与第二个 if 匹配,即该 if…else 语句是第一个 if 语句中的子句。由于执行中判断第一个 if 的表达式“a>b”为假(“3>4”为假),因此并不执行其后的 if…else 语句。这时程序将执行最下面的输出语句 printf 输出 d 值,由于 d 值一直未被改变而具有原值 2,因此最终的输出结果为 2。

13. 下面程序的功能是输出 a、b 和 c 三个变量中的最小值。请填空。

```
#include<stdio.h>
void main()
{
    int a,b,c,t1,t2;
    scanf("%d%d%d",&a,&b,&c);
    t1=a<b?____(1)____;
    t2=c<t1?____(2)____;
    printf("%d\n",t2);
}
```

【解析】

条件表达式类似于 if 语句,它是根据“?”前面子表达式的逻辑值(非 0 和 0)来判断应计算“:”前面的子表达式还是“:”后面的子表达式来作为整个条件表达式的结果。题目要求输出 a、b 和 c 三个变量中的最小值,所以当“a<b”时应选 a,否则选 b,故第(1)空应填“a:b”。同理,第(2)空应填“c:t1”。

14. 下面程序的功能是计算 $s=1+12+123+1234+12345$ 。请填空。

```
#include<stdio.h>
void main()
{
    int t=0,s=0,i;
    for(i=1;i<=5;i++)
    {
        t=i+_____ ;
        s=s+t;
    }
    printf("s=%d\n",s);
}
```

【解析】

本题要求累加一系列有规律的数。从输出结果看，s 保存的是最终结果，而 s 在循环体中每次累加一个 t，即 t 在 5 次循环中应分别等于 1、12、123、1234 和 12345。由于 t 每次循环等于“填空内容”+i，且 i 在 5 次循环中的值分别等于 1、2、3、4、5，因此在每次循环中“填空内容”应具有的值分别是 0、10、120、1230 和 12340，这一系列值正好是前一次 t 值的 10 倍，故“填空内容”应是“t*10”或者“10*t”。

15. 阅读程序，给出程序的运行结果。

```
#include<stdio.h>
void main()
{
    int k=5;
    while(--k)
        printf("%d",k-=3);
    printf("\n");
}
```

【解析】

在本题程序 while 语句的循环条件中，因为“--”运算符是前缀形式，所以表达式“--k”的值是 k 自减之后的值。程序开始时，将 k 值初始化为 5，然后进入 while 循环，此时“--k”的值为 4（非 0），所以执行循环体输出 k-=3 的值为 1，即此时的 k 值为 1。第二次执行 while 循环时，由于循环条件“--k”已使 k 值为 0，因此不再执行循环体，循环到此结束。故最终的输出结果是 1。

16. 阅读程序，给出程序的运行结果。

```
#include<stdio.h>
void main()
{
    int y=10;
    while(y--);
    printf("y=%d\n",y);
}
```

【解析】

注意，本题中 while 的循环体是一个空语句，即“(y--)”后面的那个分号“;”就是

while 语句的循环体。所以本题只要清楚表达式“y--”何时为“假”（即等于 0）就能得出结果了。由于本题中“--”运算符是后缀式，因此表达式“y--”的值是 y 自减之前的值。即当 y 等于 0 时表达式“y--”就等于 0 了，但此时由于“--”的作用，y 值还要减 1，因此循环结束后 y 的值为-1，故最终输出结果为-1。

17. 阅读程序，给出程序的运行结果。

```
#include<stdio.h>
void main()
{
    int i,j,sum;
    for(i=3;i>=1;i--)
    {
        sum=0;
        for(j=1;j<=i;j++)
            sum+=i*j;
    }
    printf("%d\n",sum);
}
```

【解析】

本题的要点是 for 循环的嵌套。外循环 for 的循环变量 i 从 3 循环递减到 1，所以外循环执行 3 次，即 i 值分别是 3、2、1。由于在外循环 for 的循环体中，每次都是给 sum 重新赋值为 0，因此只有最后一次外循环对 sum 的影响有效。故我们只考虑最后一次外循环（即 i 值为 1 时）的情况，即先给 sum 赋 0，然后进入内循环的 for 语句。内循环 for 的循环变量 j 是从 1 递增到 i，而此时 i 值为 1，即内循环只会循环一次（j 值为 1）。在内循环中，sum 被加上了 i*j 的值，而这时 i*j=1*1=1，最终 sum 的值为 1，故输出结果为 1。

18. 下面程序的功能是输出如下形式的方阵，请填空。

```
13  14  15  16
 9   10  11  12
 5   6   7   8
 1   2   3   4
```

```
#include<stdio.h>
void main()
{
    int i,j,x;
    for(j=4;j____(1)____;j--)
    {
        for(i=1;i<=4;i++)
        {
            x=(j-1)*4+____(2)____;
            printf("%4d",x);
        }
        printf("\n");
    }
}
```