第 3 章　顺序和选择结构程序设计

程序可以分为 3 种基本结构，即顺序结构、选择结构和循环结构。最简单的 C 语言程序是顺序结构程序，各语句按自上而下的顺序执行，执行完上一条语句就自动执行下一条语句，也就是说是无条件的，不必做任何判断。有时候，需要根据某个条件是否满足来决定是否执行指定的操作任务，或者从给定的两种或多种操作中选择其一，这就是选择程序结构要解决的问题。另外，在日常生活或是在程序所处理的问题中经常会遇到需要重复处理的问题，这就需要包含循环结构。顺序结构、选择结构和循环结构是结构化程序设计的 3 种基本结构，它们是各种复杂程序的基本构成单元。C 语言提供了多种语句来实现这些程序结构。

3.1　顺序结构编程

3.1.1　C 语句

一个 C 语句就是一个操作命令，通过编译后可以产生一组计算机指令，可以完成一定的操作任务。例如“c=3；”是一个赋值语句，它实现的操作是把 3 赋给变量 *c*。C 语言规定了很多语句，大致可分为以下 4 类。

1. 控制语句

控制语句用于控制程序的流程，以实现程序的各种结构方式。它们由特定的语句定义符组成。C 语言有 9 种控制语句，可分成以下三类。

(1) 条件判断语句：if 语句、switch 语句；

(2) 循环执行语句：do…while 语句、while 语句、for 语句；

(3) 转向语句：goto 语句、continue 语句、break 语句、return 语句。

2. 表达式语句

表达式语句由表达式加上分号“；”组成。其一般形式为：

表达式；

执行表达式语句也就是计算表达式的值。例如：

```
a=b+c;                          /* 赋值语句,使变量 b 加变量 c,并将所得的和赋给变量 a */
printf("I am a student.");      /* 函数调用语句,输出"I am a student."字符串 */
++i;                            /* 自增 1 语句,i 值增 1 */
sqrt(4)+3;      /* 加法运算语句,先调用 sqrt 函数,求解 4 的开平方根,再加上 3 求两值的和, */
                                /* 但计算结果不能保留,无实际意义 */
```

3. 空语句

空语句由一个分号“;”组成,是指什么也不执行的语句,常在程序中用做空循环体。例如:

```
while(getchar()!='\n');
```

该语句的功能是只要从键盘输入的字符不是回车键,则重新输入。

4. 复合语句

把多个语句用花括号{}括起来组成的一个语句称为复合语句。在程序中应把复合语句看成是单条语句。

例如:

```
{
    c=a;
    a=b;
    b=c;
}
```

注意:复合语句内的各条语句都必须以分号“;”结尾,在右括号“}”外不必再加分号。

3.1.2 顺序结构程序设计

顺序结构程序描述顺序结构的算法,其按照语句依次执行。下面介绍几个顺序结构程序设计的例子。

【例 3-1】 输入一个华氏温度 f,将其转换为摄氏温度 c 并输出。其转换公式为 $c=\frac{5}{9}\times(f-32)$。

```
#include<stdio.h>
int main()
{
    float f,c;
    printf("输入一个华氏温度: ");
    scanf("%f",&f);
    c=5.0/9*(f-32);
    printf("转换为摄氏温度为: %.2f\n",c);
    return 0;
}
```

运行结果如下:

```
输入一个华氏温度: 77
转换为摄氏温度为: 25.00
```

【例 3-2】 输入一个三位数,将该数逆序输出。例如输入 123,则输出 321。

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

```
int main()
{
    int x,a,b,c;
    printf("输入一个三位数: ");
    scanf("%d",&x);
    a=x/100;
    b=x%100/10;
    c=x%10;
    printf("逆序输出结果为: %d%d%d\n",c,b,a);
    return 0;
}
```

运行结果如下：

3.2 选择结构程序设计

C语言中提供了两种语句来实现选择结构，即if语句和switch语句。下面分别介绍这两种语句以及如何设计选择结构程序。

3.2.1 if语句

C语言提供了3种形式的if语句。

1. 单分支if语句

其形式为：

if（表达式）语句

（1）“表达式”是一个判断的条件，一般为逻辑表达式或关系表达式，同时也可以是任意的数值类型（如整型、实型、字符型、指针型数据等），此时0表示逻辑假，非0表示逻辑真。

例如：

```
① if(x>0) printf("%d为正数\n",x);                    /*表达式为条件表达式*/
② if(a>b && a>c) printf("%d为3个数中的最大数\n",a);  /*表达式为逻辑表达式*/
③ if(1) printf("Good");                               /*表达式为整型数值,表示永真*/
```

（2）“语句”由一个语句或一个复合语句组成，每个语句后面都不可缺少分号（;），若由复合语句组成则需用花括号“{}”括起。

例如：

```
if(x!=0)printf("%d为非0的值\n",x);
if(a<b)
{
    t=a;
    a=b;
```

```
    b=t;
}
```

(3) 执行过程为先判断表达式,若为真,则执行语句后再结束 if 语句;否则,直接结束 if 语句(见图 3-1)。

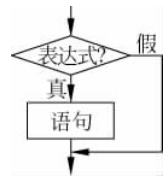

图 3-1 if 语句的执行过程

【例 3-3】 if 语句的使用:输入两个整数,输出其中的大数。

```
#include<stdio.h>
int main()
{
    int a,b,max;
    printf("Input two numbers: ");
    scanf("%d%d",&a,&b);
    max=a;
    if (max<b) max=b;
    printf("max=%d\n",max);
    return 0;
}
```

输入两个数 a 和 b,把 a 先赋给变量 max,再用 if 语句判别 max 和 b 的大小,如果 max 小于 b,则把 b 赋予 max。因此,max 中总是大数,最后输出 max 的值。运行结果如下:

```
Input two numbers: 23 32
max=32
```

2. 双分支 if 语句

其形式为:

if (表达式) 语句 1 else 语句 2

(1) "表达式"、"语句 1"和"语句 2"同上。例如:

```
if(x>y)max=x; else max=y;
if(a<0)b=-1;  else b=0;
```

(2) 执行过程为先判断表达式,若为真,则执行语句 1 后再结束 if 语句;否则,执行语句 2 后再结束 if 语句(见图 3-2)。

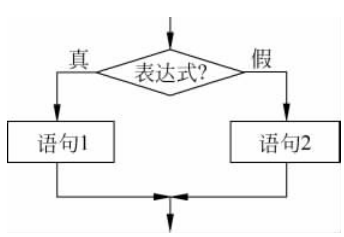

图 3-2 双分支 if 语句的执行过程

【例 3-4】 if…else 语句的使用:输入两个整数,输出其中的大数。

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

```
{
    int a, b;
    printf("input two numbers: ");
    scanf("%d%d",&a,&b);
    if(a>b)
        printf("max=%d\n",a);
    else
        printf("max=%d\n",b);
    return 0;
}
```

程序的运行结果为：

```
input two numbers: 23 32
max=32
```

3. 多分支 if 语句

其形式为：

if（表达式 1）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 m）语句 m
else 语句 m+1

（1）“表达式 i”和“语句 i”($0<i<m+2$)同上说明。

例如有一个函数：

$$y=\begin{cases}-1 & (x<0)\\ 0 & (x=0)\\ 1 & (x<0)\end{cases}$$

```
if (x<0) y=-1;
else if(x>0)y=1;
else y=0;
```

（2）执行过程。

S_1：判断表达式 1，若为真，则执行语句 1 后结束 if 语句；否则，执行 S_2；

S_2：判断表达式 2，若为真，则执行语句 2 后结束 if 语句；否则，执行 S_3；

S_3：判断表达式 3，若为真，则执行语句 3 后结束 if 语句；否则，执行 S_4；

⋮

S_i：同上判断表达式 i($3<i<m$)，若为真，则执行语句 i 后结束 if 语句；否则，执行 S_{i+1}；

⋮

S_m：判断表达式 m，若为真，则执行语句 m 后结束 if 语句；否则，执行 S_{m+1}；

S_{m+1}：执行语句 $m+1$ 后结束 if 语句(见图 3-3)。

由图 3-3 可知：只有当表达式 1 到表达式 $i-1$ 都为假时才判断表达式 i；所有的语句中只有一个语句被执行；若执行语句 i，则条件为表达式 1 到表达式 $i-1$ 都为假且表达式 i 为真。

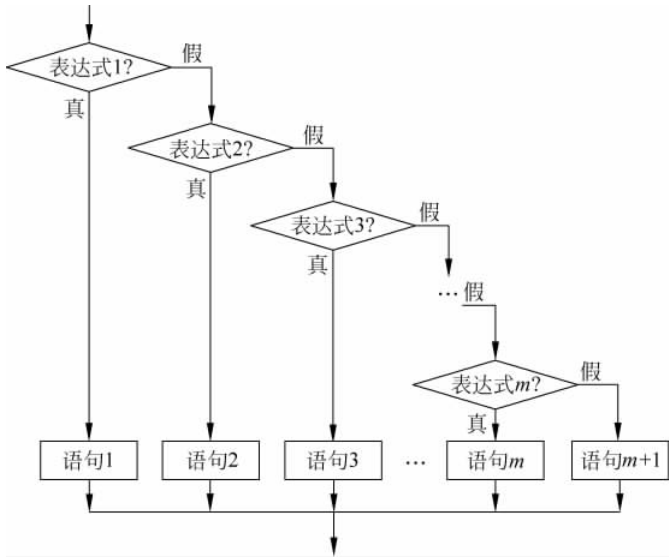

图 3-3　多分支 if 语句的执行过程

4. if 语句的嵌套

if 语句的嵌套就是指在 if 语句中又包含一个或多个 if 语句。例如：

(1)

```
if (表达式 1)
    if (表达式 2) 语句 1
    else 语句 2
else
    if(表达式 3) 语句 3
    else 语句 4
```

(2)

```
if (表达式 1) 语句 1
else
    if (表达式 2) 语句 2
    else 语句 3
```

说明：

(1) if 的个数一定大于或等于 else 的个数。

(2) if 和 else 的配对原则：从第一个 else 开始，向上查找，最近的未配对的 if 与之配对，再用同样的方法依次查找下一个 else 配对的 if。

(3) 有时为了表示语句的逻辑关系，可以加花括号来确定配对关系。例如：

①

```
if (x>=0)
    if (x>0) y=1
    else y=0;
```

表示为：

$$y=\begin{cases}0 & (x=0)\\ 1 & (x>0)\end{cases}$$

②

```
if (x>=0)
{ if (x>0) y=1;}
else y=0;
```

表示为：

$$y=\begin{cases}0 & (x<0)\\ 1 & (x>0)\end{cases}$$

【例 3-5】 输入一个年份，判定其是否为闰年。

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

```
int main()
{
    int y,flag;
    printf("Please input a year: ");
    scanf("%d",&y);
    if(y%4!=0)
        flag=0;
    else if(y%100!=0)
        flag=1;
    else if(y%400!=0)
        flag=0;
    else flag=1;
    if(flag)
        printf("%d is a leap.\n",y);
    else
        printf("%d is not a leap.\n",y);
    return 0;
}
```

运行结果如下：

```
Please input a year: 2014
2014 is not a leap.
```

3.2.2 条件运算符

条件运算符是C语言中唯一的一个三目运算符，包括3个操作对象。它求解的过程和双分支if语句的求解过程很相似。

其一般形式为：

表达式1? 表达式2:表达式3

例如：

```
max= a>b? a:b;          /* max得到a、b中的最大值 */
```

说明：

(1) 条件运算符的优先级为13级，高于逗号运算符和赋值运算符的优先级。

(2) 条件运算符的结合方向为“自右向左”。

假设 $x=0$、$y=1$、$a=3$、$b=2$，则 $x>y?\ x:a>b?\ a:b$ 表达式的值为多少？

该表达式相当于 $x>y?\ x:(a>b?\ a:b)$，所以结果为3。

(3) 执行顺序为先求解表达式1，若为非0(真)则求解表达式2，此时表达式2的值作为整个条件表达式的值；若为0(假)则求解表达式3，此时表达式3的值作为整个条件表达式的值。例如：

```
char c;
c=(c>=97 && c<=122)?(c-32):c; /* c若为小写字母则转换为大写,若为大写字母不转换 */
```

(4) 在条件表达式中，表达式1、表达式2和表达式3的类型可不同(但要兼容)。例如：

```
int x=1,y=2,t;
t=x>y? 'a':'b';
```

关系表达式 $x>y$ 的值为 0,表达式 1 为整型值,而'a'和'b'为字符型常量。

【例 3-6】 条件运算符举例。

```
#include<stdio.h>
int main()
{
    char c='k';
    int i=1, j=2, k=3;
    float x=3e+5, y=0.85;
    printf("%d,%d\n",!x*!y,!!!x);
    printf("%d,%d\n",x||i&&j-3,i<j&&x<y);
    printf("%d,%d\n",i==5&&c&&(j=8),x+y||i+j+k);
    return 0;
}
```

在本例中,$!x$ 和$!y$ 都为 0,$!x*!y$ 也为 0,故其输出值为 0。由于 x 为非 0,故$!!!x$ 的逻辑值为 0。对于 $x||i\&\&j-3$,先计算 $j-3$ 的值为非 0,再求 $i\&\&j-3$ 的逻辑值为 1,故 $x||i\&\&j-3$ 的逻辑值为 1。对于 $i<j\&\&x<y$,由于 $i<j$ 的值为 1,而 $x<y$ 为 0,故表达式的值为 1、0 相与,最后为 0。对于 $i==5\&\&c\&\&(j=8)$,由于 $i==5$ 为假,即值为 0,该表达式由两个与运算组成,所以整个表达式的值为 0。对于 $x+y||i+j+k$,由于 $x+y$ 的值为非 0,故整个或表达式的值为 1。运行结果如下:

```
0,0
1,0
0,1
```

【例 3-7】 编写程序,输入一个字符,判断其是否为大写字母,如果是大写字母则将其转换为小写字母,否则不变。

使用 if 语句编写的程序如下:

```
#include<stdio.h>
int main()
{
    char ch;
    printf("Please input ch:");
    scanf("%c",&ch);
    if (ch>='A'&&ch<='Z')
        ch=ch+32;
    printf("ch=%c\n",ch);
    return 0;
}
```

使用条件运算符编写的程序如下:

```
#include<stdio.h>
int main()
{
    char ch;
    printf("Please input ch:");
```

```
    scanf("%c",&ch);
    ch=(ch>='A'&&ch<='Z')?ch+32:ch;
    printf("ch=%c\n",ch);
    return 0;
}
```

程序的运行结果如下：

```
Please input ch:D
ch=d
```

3.2.3 switch 语句

switch 语句是另外一种选择语句，它是一种实现多分支的选择语句。在实际问题中经常需要用到多分支选择，可以用嵌套的 if 语句来处理，但有时会显得分支较多，程序冗长，而用 switch 语句处理会更直接、更简单。

其一般形式为：

```
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
}
```

例如：成绩 mark 分等级（90 分或 90 分以上输出 A，80～89 输出 B，70～79 输出 C，60～69 输出 D，60 分以下输出 E）。

```
switch ((int)(mark/10))
{
    case  10: printf("A");
    case  9: printf("A");
    case  8: printf("B");
    case  7: printf("C");
    case  6: printf("D");
    default: printf("E");
}
```

（1）switch 的后面为一个表达式，其值可以为任何类型。

（2）case 后面的表达式一定是常量表达式，其值的类型一般为整型或可自动转为整型的类型（例如字符型）；每个常量表达式的值必须不相同。

（3）“常量表达式 i”（$1 \leqslant i \leqslant n$）只是起语句标号作用，就像一个“入口点”，并不是在此处进行条件判断。

（4）语句 i（$1 \leqslant i \leqslant$ n）可以由一个语句或多个语句组成；每个语句后都不可缺少分号（;）。当由多个语句组成时，可以不必用花括号括起，系统会自动顺序地执行这些语句；当然，加花括号也可以，把其视为一个复合语句。

(5) 各个 case 和 default 出现的次序可以随意。上例可写成:

```
switch ((int)(mark/10))
{
    case 7: printf("C");
    case 10: printf("A");
    case 9: printf("A");
    case 8: printf("B");
    default : printf("E");
    case 6: printf("D");
}
```

(6) 执行过程如下。

S_1:求解表达式;

S_2:查找“入口点”,从第一个 case 开始,若该 case 后的常量表达式的值和表达式的值相同,则第一个 case 为“入口点”,若不相同则判断下一个 case,依次判断下去,直到查找到“入口点”,若所有 case 都不为“入口点”,则 default 为“入口点”;

S_3:从“入口点”开始,依次执行下面的每条语句(包括“入口点”的语句)。

【例 3-8】 switch 语句实现成绩分等级。

```
#include<stdio.h>
int main()
{
    float mark;
    printf("Please input a mark:");
    scanf("%f",&mark);
    switch ((int)(mark/10))
    {
    case 7: printf("C");
    case 10: printf("A");
    case 9: printf("A");
    case 8: printf("B");
    default: printf("E");
    case 6: printf("D");
    }
    printf("\n");
    return 0;
}
```

运行结果如下:

```
Please input a mark:90.5
ABED
Please input a mark:48.5
ED
```

(7) 通常,在执行了一个 case 分支后就应该结束 switch 语句,例如上例中的 mark 为 90.5,只需要输出 A 即可。为此,可以在每个语句后添加 break 语句来达到此目的(break 语句有终止 switch 语句执行的作用)。

因此,上例可改为:

```
#include<stdio.h>
int main()
{
    float mark;
    printf("Please input a mark:");
    scanf("%f",&mark);
    switch ((int)(mark/10))
    {
    case 7: printf("C"); break;
    case 10: printf("A"); break;
    case 9: printf("A"); break;
    case 8: printf("B"); break;
    default: printf("E"); break;
    case 6: printf("D"); break;
    }
    printf("\n");
    return 0;
}
```

运行结果如下：

```
Please input a mark:90.5
A
```

```
Please input a mark:48.5
E
```

(8) 多个 case 可以共用一组执行语句，例如：

```
#include<stdio.h>
int main()
{
    float mark;
    printf("Please input a mark:");
    scanf("%f",&mark);
    switch ((int)(mark/10))
    {
    case 10:                /*可看作一个空语句*/
    case 9: printf("A"); break;
    case 8: printf("B"); break;
    case 7: printf("C"); break;
    case 6: printf("D"); break;
    default: printf("E"); break;
    }
    printf("\n");
    return 0;
}
```

运行结果如下：

```
Please input a mark:100
A
```

3.2.4 选择结构程序设计举例

【例 3-9】 求一元二次方程 $ax^2+bx+c=0$ 的根。

该一元二次方程的根有以下 4 种情况：

(1) 若 $a=0$，不是二次方程。

(2) 若 $b^2-4ac=0$，有两个相同的实根。

(3) 若 $b^2-4ac>0$，有两个不同的实根。

(4) 若 $b^2-4ac<0$，有两个共轭复根。

```
#include<math.h>
#include<stdio.h>
int main()
{
    float a,b,c,disc,x1,x2,f1,f2;
    printf("Input a,b,c:");
    scanf("%f,%f,%f",&a,&b,&c);
    if(fabs(a)<=1e-6)
        printf("It is not a quadratic.");
    else
        disc=b*b -4*a*c;
    if( fabs(disc)<= 1e-6)
        printf("It has two equal roots:%8.2f\n", -b/(2*a));
    else if(disc>1e-6)
    {
        x1=(-b+sqrt(disc))/(2*a);
        x2=(-b-sqrt(disc))/(2*a);
        printf("It has two real roots:%8.2f and %8.2f\n",x1,x2);
    }
    else
    {
        f1=-b/(2*a);
        f2=sqrt(-disc)/(2*a);
        printf("It has two complex roots:%8.2f+%.2fi",f1,f2);
        printf("and %8.2f-%.2fi\n",f1,f2);
    }
    return 0;
}
```

运行结果如下：

```
Input a,b,c:1,2,1
It has two equal roots:   -1.00
```

```
Input a,b,c:3,5,1
It has two real roots:   -0.23 and    -1.43
```

```
Input a,b,c:2,3,4
It has two complex roots:   -0.75+1.20i and    -0.75-1.20i
```

【例 3-10】 某市电费的收费标准如下：用电量小于 30 度，收费 6 元；30～60 度，电费按每度 1.8 元计算；60～120 度，电费按每度 1.6 元计算；120 度以上，电费按每度 1.4 元计

算。输入用电量，计算出电费为多少。

```
#include<stdio.h>
int main()
{
    float s,f;
    printf("Input the power:");
    scanf("%f",&s);
    switch ( (int)(s)/30)
    {
    case 0: f=6; break;
    case 1: f=6+(s-30)*1.8; break;
    case 2:
    case 3: f=6+30*1.8+(s-60)*1.6; break;
    default: f=6+30*1.8+60*1.6+(s-120)*1.4; break;
    }
    printf("The expense of the power is %.2f.\n",f);
    return 0;
}
```

运行结果如下：

```
Input the power:20
The expense of the power is 6.00.

Input the power:50
The expense of the power is 42.00.

Input the power:100
The expense of the power is 124.00.

Input the power:268
The expense of the power is 363.20.
```

3.3 本章小结

一、知识点小结

内　　容	描　　述	备　　注
关系运算符	>、<、>=、<=、==、!=	>、<、>=、<=的优先级别高于==、!= 关系运算符的优先级别低于算术运算符
逻辑运算符	&&、‖、!	除逻辑非为一元运算符以外，&& 和 ‖ 均为二元运算符，并且！的优先级别高于 &&，&& 高于 ‖
条件运算符	?:	三元运算符
if 形式的条件语句	if（表达式）语句 A	用于单分支选择控制
if…else 形式的条件语句	if（表达式）语句 1 else 语句 2	用于双分支选择控制

续表

内　　容	描　　述	备　　注
else…if 形式的条件语句	if (表达式) 语句 1 else if (表达式 2) 语句 2 ⋮ else if(表达式 m) 语句 m else 语句 m+1	用于多分支选择控制
switch 语句	switch(表达式) { case 常量 1: 语句序列 1 case 常量 2: 语句序列 2 ⋮ case 常量 n: 语句序列 n default: 语句序列 n+1	用于多分支选择控制

二、常见错误小结

常见错误实例	错误描述	错误类型
if(a>b); 　max=a	在紧跟着 if 单分支选择语句的条件表达式的圆括号之后写了一个分号	运行时错误

if(a>b);
　max=a;
else
　max=b;
	在紧跟着 if…else 双分支选择语句的条件表达式的圆括号之后写了一个分号	编译错误

if(a>b)
　max=a;
printf("max=%d \n",a);
	在界定 if 语句后的复合语句时忘了写花括号	运行时错误

if(a>b)
　max=a;
　printf("max=%d \n",a);
else
　max=b;
　printf("max=%d \n",b);
	在界定 if…else 语句后的复合语句时忘了写花括号。由于 if 或 else 子句中只允许有一条语句,因此在需要多条语句时必须用复合语句,即把需要执行的多条语句用一对花括号括起来	编译错误

	将关系运算符与相应的数学运算符混淆,写成≠、≤、≥	无法输入
if(a= =b) 　printf("a=b \n");	在关系运算符>=、<=、==、!=的中间加入空格	编译错误
if(a=b) 　printf("a=b \n");	在 if 语句的条件表达式中表示相等条件时,将关系运算符==误用为赋值运算符	运行时错误
if(a =<b) printf("max=%d \n",b);	将关系运算符!=、<=、>=的两个符号写反,写成了=!、=<、=>	编译错误
if(x==1.1)	用==或者!=测试两个浮点数是否相等,或者判断一个浮点数是否等于 0	运行时错误

续表

常见错误实例	错误描述	错误类型
if('A'<=ch<='Z')	误以为语法上合法的关系表达式在逻辑上一定是正确的	运行时错误

switch(mark)
{
case 10:
case 9:printf("A \n");
case 8:printf("B \n");
⋮
}
	在 switch 语句中需要每个 case 分支单独处理时，缺少 break 语句	运行时错误

switch(mark)
{
case10:
case9:printf("A \n");
break;
case8:printf("B \n");
break;
⋮
}
	在 switch 语句中，case 和其后的数值常量中间缺少空格	运行时错误

switch(mark)
{
case 100:
case 90～100:printf("A \n");
break;
case mark<90:printf("B \n");
break;
⋮
}
	在 switch 语句中，case 后的常量表达式用一个区间表示，或者出现了运算符(例如关系运算符)	编译错误

习题 3

一、基础题

1. 设 a、b 和 c 都是 int 型变量，且 $a=3$、$b=4$、$c=5$，则下面的表达式中值为 0 的表达式是_____。

A) 'a'&&'b'　　B) $a<=b$

C) $a||b+c\&\&b-c$　　D) !((a<b)&&! c||1)

2. 设 x、y、z、t 均为 int 型变量，则执行以下语句后 t 的值为_____。

```
x=y=z=1;
t=++x || ++y && ++z;
```

A) 不定值　　B) 4　　C) 1　　D) 0

3. 若运行下面的程序时给变量 a 输入 15,则输出结果是_____。

```
#include<stdio.h>
int main()
{
    int a, b;
    scanf("%d", &a);
    b=a>15?a+10:a-10;
    printf("%d\n", b) ;
    return 0;
}
```

A) 5　　B) 25　　C) 15　　D) 10

4. 以下程序的运行结果是_____。

```
#include<stdio.h>
int main()
{
    int a=0, b=1, c=0, d=20, x;
    if (a) d=d-10;
    else if (!b)
        if (!c) x=15;
        else x=25;
    printf("%d\n", d);
    return 0;
}
```

A) 15　　B) 25　　C) 20　　D) 10

5. 若 k 是 int 型变量,则下面程序片段的输出结果是_____。

```
k=8;
if (k<=0)
    if (k==0) printf("####")
    else printf("&&&&");
else printf("****");
```

A) ####　　B) &&&&

C) ****　　D) 有语法错误,无输出结果

6. 假定所有变量均已正确说明,下列程序段运行后 x 的值是_____。

```
a=b=c=0; x=35;
if (!a) x--;
    else if (b);
if (c) x=3;
    else x=4;
```

A) 34　　B) 4　　C) 35　　D) 3

7. 设 ch 是 char 型变量,其值为 A,则下面的表达式的值是_____。

```
ch=(ch>='A'&&ch<='Z')?(ch+32):ch
```

A) A　　B) a　　C) Z　　D) z

8. 下列能正确表示 a 和 b 同时为正或同时为负的逻辑表达式的是_____。

A) (a>=0 || b>=0) && (a<0 || b<0)

B) (a>=0 && b>=0) && (a<0 && b<0)

C) (a+b>0 && a+b<=0)

D) a*b>0

9. 为了避免嵌套 if…else 语句的二义性,C 语言规定 else 总是与_____组成配对关系。

A) 缩排位置相同的 if　　B) 在其之前未配对的 if

C) 在其之前未配对的最近的 if　　D) 同一行上的 if

10. 在运行下面的程序时从键盘输入"1605<回车>",则输出结果是_____。

```
#include<stdio.h>
int main()
{
    int t,h,m;
    scanf("%d",&t);
    h=(t/100)%12;
    if (h==0) h=12;
    printf("%d:",h);
    m=t%100;
    if (m<10) printf("0");
    printf("%d",m);
    if (t<1200||t==2400)
        printf("AM");
    else printf("PM");
    printf("\n");
    return 0;
}
```

A) 6:05PM　　B) 4:05PM　　C) 16:05AM　　D) 12:05AM

11. 下面程序的输出结果是_____。

```
#include<stdio.h>
int main()
{
    int a=2, b=7, c=5;
    switch(a>0)
    {
    case 1:
        switch(b<0)
        {
        case 1: printf("@"); break;
        case 0: printf("!"); break;
        }
    case 0:
        switch(c==5)
        {
        case 0: printf("*"); break;
```

```
            case 1: printf("#"); break;
            default: printf("%%");break;
            }
        default: printf("&");
        }
        printf("\n");
        return 0;
    }
```

A) &　　B) !#&　　C) %%　　D) @*&

12. 在运行下面的程序时从键盘输入数据为"2,13,5<回车>",则输出结果是_____。

```
#include <stdio.h>
int main()
{
    int a,b,c;
    scanf("%d,%d,%d",&a,&b,&c);
    switch(a)
    {
    case 1: printf("%d\n",b+c); break;
    case 2: printf("%d\n",b-c); break;
    case 3: printf("%d\n",b*c); break;
    case 4: { if(c!=0) {printf("%d\n",b/c);break;}
              else {printf("error\n");break;}
            }
    defualt: break;
    }
    return 0;
}
```

A) 10　　B) 8　　C) 65　　D) error

二、程序设计题

1. 输入圆的半径 r,求该圆的周长、面积,结果保留两位小数。

2. 编写一个程序,功能是输入一个整数,将它反向输出。例如输入 12345,则输出 54321。

3. 从键盘输入你和你朋友的年龄,编程判断谁的年龄最大,并打印最大者的年龄。

4. 编写程序,输入一个字符,判断该字符是大写字母、小写字母、数字还是其他字符。

5. 编写程序,输入三位整数,判断该数除以 9 的商是否等于它各位数字的平方和。例如 224,它除以 9 的商为 24,而 $2^2+2^2+4^2=24$。

6. 编写程序,判断两位整数 m 是否为守形数。守形数是指该数本身等于自身平方的低位数,例如 25 是守形数,因为 $25^2=625$,而 625 的低两位是 25。

7. 判断某人是否属于肥胖体型。根据身高与体重因素,医务工作者经过广泛的调查分析给出了以下按"体指数 t"对肥胖程度的划分:

体指数 $t=w/h^2$　(w 为体重,单位为 kg; h 为身高,单位为 m)

(1) 当 $t<18$ 时,为低体重;

(2) 当 t 介于 18 和 25 之间时,为正常体重;

(3) 当 t 介于 25 和 27 之间时,为超重体重;

(4) 当 $t \geqslant 27$ 时,为肥胖。

编程从键盘输入你的体重 w 和身高 h,根据给定公式计算出体指数 t,然后判断你的体重属于何种类型。

8. 假设某地的个人所得税的起征额为 1600 元,若超过 1600 元,纳税额按照以下方法计算:

(1) 超过 500 元以内部分,税率 5%;

(2) 超过 500 至 2000 元部分,税率 10%;

(3) 超过 2000 至 5000 元部分,税率 15%;

(4) 超过 5000 至 20 000 元部分,税率 20%;

(5) 超过 20 000 至 40 000 元部分,税率 25%;

(6) 超过 40 000 至 60 000 元部分,税率 30%;

(7) 超过 60 000 至 80 000 元部分,税率 35%;

(8) 超过 80 000 至 100 000 元部分,税率 40%;

(9) 超过 100 000 元部分,税率 45%。

编写个人所得税计算器,输入个人月收入总额,计算应纳个人所得税金额。

9. 编程设计一个简单的猜数游戏:先由计算机"想"一个数请人猜,如果人猜对了,则计算机给出提示"Right!",否则提示"Wrong!",并告诉人所猜的数是大还是小。提示:计算机"想"一个数可以用随机函数 rand()产生。随机函数 rand()产生一个 0～RAND_MAX 的整数,RAND_MAX 是在头文件 stdlib.h 中定义的符号常量,因此使用该函数时需要包含头文件 stdlib.h。ANSI 标准规定 RAND_MAX 的值不得大于双字节整数的最大值 32 767。

10. 编程设计一个简单的计算器程序,要求根据用户从键盘输入的表达式:

操作数 1　运算符 op　操作数 2

计算表达式的值,指定的运算符为加(+)、减(-)、乘(*)、除(/)。