

第 4 章 选择结构程序设计

实际生活中，若需要处理事情，往往会根据不同的情况，采取相应的不同处理措施。在程序设计中，这种算法表现为根据不同的判定条件控制执行不同的程序流程。选择结构是程序结构中很重要的一种控制结构。在程序设计时，当需要进行选择、判断和处理的时候，就要用到选择结构。C 语言中设计选择结构的语句一般包括单 if 语句、if-else 语句、if-else if-else 语句和 switch 语句等，下面分别加以介绍。

4.1 if 语句

C 语言中 if 语句是常用的条件判断语句，用来判定是否满足指定的条件，并根据判定结果执行相应的操作。C 语言提供了三种形式的 if 语句，在使用时可根据具体问题的复杂程度选择合适的形式。

4.1.1 if 语句的形式

1. 单 if 语句

其一般形式为：

if(表达式) 语句

这种形式的 if 语句的执行过程如图 4-1 所示。系统首先对表达式求解，当结果为“真”(非 0) 时，则执行指定的语句；否则跳过指定语句，接着执行 if 语句的下一语句。

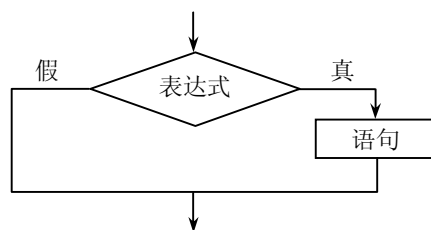


图 4-1 if 语句执行过程

【例 4.1】 求两个整数中较大者。

分析：可设三个变量 a, b, max。首先假设 max=a，然后判断 max 和 b，若 max<b，则 max=b，最后输出 max。

程序如下：

```
#include "stdio.h"
void main()
{
    int a,b,max;
    printf("请输入 a,b 的值\n");
```

```
scanf("%d,%d",&a,&b);  
max=a;  
if(max<b) max=b;  
printf("max=%d",max);  
}
```

2. if-else 语句

其一般形式为:

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

执行过程如图 4-2 所示。系统先对表达式求解,当结果为“真”(非 0)时,执行语句 1;当结果为“假”(0)时,执行语句 2。

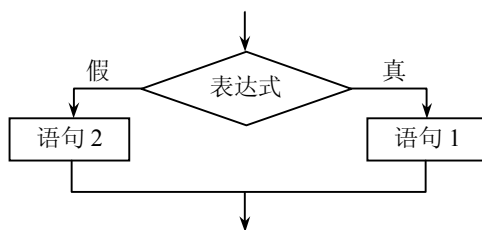


图 4-2 if-else 语句执行过程

【例 4.2】 利用 if-else 语句编写程序,求两个整数中较大者。

```
#include "stdio.h"  
void main()  
{  
    int a,b;  
    printf("a,b=");  
    scanf("%d,%d",&a,&b);  
    if (a>b)  
        printf("max=%d\n",a);  
    else  
        printf("max=%d\n",b);  
}
```

3. if-else if-else 语句

前两种形式的 if 语句一般都用于两个分支的情况。当有多个分支选择时,可采用 if-else if-else 语句,其一般形式为:

```
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 n-1)
    语句 n-1
else
    语句 n

```

执行过程如图 4-3 所示。系统首先求解表达式 1 的值，当其值为“真”（非 0）时，执行语句 1，然后跳出该选择结构；否则求解表达式 2 的值，当其值为“真”（非 0）时，则执行语句 2，然后跳出该选择结构；否则再继续求解表达式 3，当其值为“真”（非 0）时，执行语句 3，然后跳出该选择结构；否则接着进行判断，依次类推。如果经过求解，所有的表达式都为“假”（0），就执行最后一个 else 部分的语句 n，从而结束整个 if 语句。

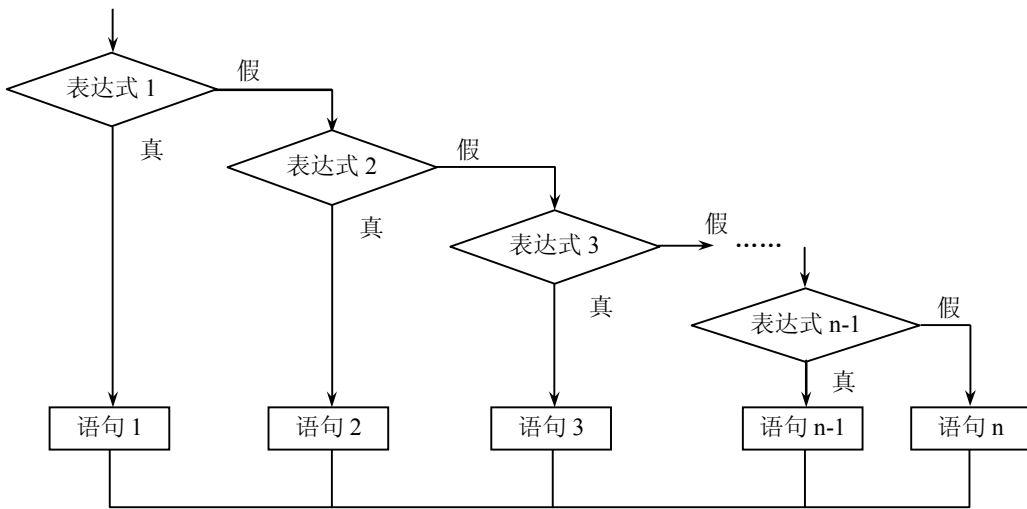


图 4-3 if-else if-else 语句执行过程

【例 4.3】 给出百分制的成绩，要求输出成绩等级 A、B、C、D、E。90 分以上为 A，80~89 分为 B，70~79 分为 C，60~69 分为 D，60 分以下为 E。

分析：设成绩为 score，假设取值在 0~100 之间，没有错误的输入，则有：

```

60 > score ≥ 0      E
69 ≥ score ≥ 60    D
79 ≥ score ≥ 70    C
89 ≥ score ≥ 80    B
100 ≥ score ≥ 90   A

```

程序如下：

```

#include "stdio.h"
void main()
{
    int score;
    printf("请输入学生的成绩\n");
    scanf("%d",&score);
    if (score>=0 && score<60)

```

```
    printf("grade is E\n");
else if (score>=60 && score<=69)
    printf("grade is D\n");
else if (score>=70 && score<=79)
    printf("grade is C\n");
else if (score>=80 && score<=89)
    printf("grade is B\n");
else
    printf("grade is A\n");
}
```

这里有两个问题，需要我们共同探讨。

问题 1：条件表达式可否采用更简洁的形式？

由于本题假设没有错误的输入，成绩 `score` 是在 `0~100` 之间的，而且我们知道 `if-else if-else` 语句执行时，下一个条件的判断是在上一个条件为“假”时进行的。以第二个条件为例，当我们对 `(score>=60 && score<=69)` 进行判断时，实际 `score>=60` 是必然的，这可由第一个条件 `(score>=0 && score<60)` 为“假”知道，所以 `score>=60` 可以省略。

那么程序可修改如下：

```
#include "stdio.h"
void main()
{
    int score;
    printf("请输入学生的成绩\n");
    scanf("%d",&score);
    if (score>=0 && score<60)
        printf("grade is E\n");
    else if (score<=69)
        printf("grade is D\n");
        else if (score<=79)
            printf("grade is C\n");
            else if (score<=89)
                printf("grade is B\n");
                else
                    printf("grade is A\n");
}
```

问题 2：条件表达式的先后可否任意？

若将第二个条件移到后面最后一个条件前，则程序如下：

```
#include "stdio.h"
void main()
{
    int score;
    printf("请输入学生的成绩\n");
    scanf("%d",&score);
    if (score>=0 && score<60)
        printf("grade is E\n");
    else if (score<=79)
        printf("grade is C\n");
```

```
        else if (score<=89)
printf("grade is B\n");
        else if (score<=69)
printf("grade is D\n");
        else
printf("grade is A\n");
}
```

输入 65 后，程序运行结果如下：

```
score=65
grade is C
```

分析：当从键盘上输入 65 时，输出 grade is C，得出了错误结果。这是因为，当从键盘上输入 65 时，score=65，第一个条件为“假”，接着判断第二个条件（score<=79），条件为“真”，输出 grade is C，而不会判断到第三个条件（score<=69）。

所以在采用 if-else if-else 语句处理多分支时，要注意条件的表示形式和先后顺序。

在使用 if 语句中还应注意以下问题：

(1) if 语句中的表达式通常是逻辑表达式或关系表达式，也可以是其他类型的表达式，如赋值表达式等，甚至可以是一个变量。

例如：

if(a=5) 语句;

if(b) 语句;

都是允许的。只要表达式的值为非 0，即为“真”；为 0，即为“假”。

又如，有程序段：

```
if (a=b)
printf("%d",a);
else
printf("a=0");
```

该语句的语义是：把 b 值赋予 a，如为非 0 则输出该值，否则输出"a=0"字符串。这种用法在程序中是经常出现的。

(2) 在 if 语句中，表达式必须用括号括起来，在语句之后必须加分号。

(3) 在 if 语句的三种形式中，所有的语句应为单个语句，如果要想在满足条件时执行一组（多个）语句，则必须把这一组语句用 {} 括起来组成一个复合语句，但要注意的是在 {} 之后不能再加分号。

例如：

```
if (a>b)
{a++;
b++;}
else
{a=0;
b=10;}
```

4.1.2 if 语句的嵌套

在 if 语句中又包含一个或多个 if 语句称为 if 语句的嵌套。

if 语句可以内嵌在 if 子句中，又可以内嵌在 else 子句中。例如：

```

if ( )
    if ( )
        语句 1
    else
        语句 2
else
    if ( )
        语句 3
    else
        语句 4

```

内嵌 if

内嵌 if

应当注意 if 与 else 的配对关系。else 总是与它上面最近的 if 配对。假如写成：

```

if ( )
    if ( ) 语句 1
else
    if ( )
        语句 2
    else
        语句 3

```

编程者把 else 写在与第一个 if (外层 if) 同一列上，希望 else 与第一个 if 对应，但实际上 else 是与第二个 if 配对，因为它们相距最近。因此最好使内嵌 if 语句也包含 else 部分，这样 if 的数目和 else 的数目相同，从内层到外层一一对应，不致出错。

如果 if 与 else 的数目不一样，为实现程序设计者的企图，可以加花括号来确定配对关系。例如：

```

if ( )
    {if ( ) 语句 1}      (内嵌 if)
else
    语句 2

```

这时 { } 限制了内嵌 if 语句的范围，因此 else 与第一个 if 配对。

【例 4.4】 猜数程序，和预定的数比较，并给出相应的提示。

```

#include "stdio.h"
void main()
{
    int number=200;
    int i;
    scanf("%d",&i);
    if (i>number)
        printf("猜大了!");
    else
        if (i<number)
            printf("猜小了!");
    else

```

```
        printf("猜对了!");  
    }
```

【例 4.5】 求三个整数 a, b, c 中的最大数。

```
#include "stdio.h"  
void main()  
{  
    int a,b,c,max;  
    printf("a,b,c=");  
    scanf("%d,%d,%d",&a,&b,&c);  
    max=a;  
    if (b>c)  
        {if (b>a)  
            max=b;}  
    else  
        if (c>a)  
            max=c;  
    printf("max=%d\n",max);  
}
```

为了实现 else 与第一个 if 的配对, 应该用花括弧将第二个 if 语句括起来; 否则, 尽管根据缩进格式, 我们认为 else 将与第一个 if 配对, 但实际上 else 与第二个 if 配对。

通过这个例子可以认识到:

(1) 不平衡的 if-else 结构会增加阅读和理解程序的困难。

(2) 正确的缩进格式 (即锯齿形书写格式) 可以帮助人们理解程序, 但错误的缩进格式反而会使人迷惑。

(3) 不要太相信自己的判断, 要严格按语法关系检查程序。在不易弄清的地方可以加花括号来保证自己构思的逻辑关系的正确性。

本例采用 if 语句的嵌套结构来实现, 实质上是为了进行多分支选择。此类问题用 if-else if-else 语句也可以实现, 而且程序更加清晰, 便于阅读理解。因此, 在一般情况下, 多分支问题较少使用 if 语句的嵌套结构来实现。

下面采用 if-else if-else 语句改写例 4.5, 程序如下:

```
#include "stdio.h"  
void main()  
{  
    int a,b,c,max;  
    printf("请输入 a,b,c 的值\n");  
    scanf("%d,%d,%d",&a,&b,&c);  
    if (a>b && b>c)  
        max=a;  
    else if (b>c && b>a)  
        max=b;  
    else  
        max=c;  
}
```

```
    printf("max=%d\n",max);  
}
```

4.2 switch 语句

C 语言还提供了另一种用于多分支选择的 switch 语句，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;]  
}
```

其中，表达式可以为任何类型，各个常量表达式代表 switch 后面的表达式的各个不同的取值。执行时，系统首先求解表达式的值，然后依次与各个 case 后面给出的常量表达式的值相比较，当表达式的值与某个 case 后的常量表达式的值相等时，就从此处开始执行该 case 后面的语句，而不再进行判断。如果所有的 case 后面的常量表达式的值都和表达式的值不匹配，系统就执行 default 后面的语句，如果程序省略了 default 语句，那么将不作任何处理，接着执行该选择结构下面的语句。

【例 4.6】 输入一个星期数字，输出其对应的英文单词。

```
#include "stdio.h"  
void main()  
{  
    int a;  
    printf("请输入一个星期数字(1-7):");  
    scanf("%d",&a);  
    switch (a)  
    {  
        case 1:printf("Monday\n");  
        case 2:printf("Tuesday\n");  
        case 3:printf("Wednesday\n");  
        case 4:printf("Thursday\n");  
        case 5:printf("Friday\n");  
        case 6:printf("Saturday\n");  
        case 7:printf("Sunday\n");  
        default:printf("Error!\n");  
    }  
}
```

本程序是要求输入一个星期数字，输出其对应的英文单词。但是当输入 3 之后，却执行了 case 3 以及以后的所有语句，输出了 Wednesday 及以后的所有单词。这当然不是希望的。为什么会出现这种情况呢？

这说明了 switch 语句的执行特点。在 switch 语句中，“case 常量表达式”只相当于一个语

句标号,当表达式的值和某标号相等则转向该标号执行,但不能在执行完该标号的语句后自动跳出整个 switch 语句,所以出现了继续执行所有后面 case 语句的情况。这与前面介绍的 if 语句是完全不同的,应特别注意。

为了避免上述情况,C 语言还提供了一种 break 语句,专用于跳出 switch 语句,break 语句只有关键字 break,没有参数。修改例题程序,在每一 case 语句之后增加 break 语句,使每一次执行之后均可跳出 switch 语句,从而避免输出不应有的结果。

修改例 4.6 中的 switch 语句,添加了 break 语句后为:

```
switch (a)
{
    case 1:printf("Monday\n");break;
    case 2:printf("Tuesday\n");break;
    case 3:printf("Wednesday\n");break;
    case 4:printf("Thursday\n");break;
    case 5:printf("Friday\n");break;
    case 6:printf("Saturday\n");break;
    case 7:printf("Sunday\n");break;
    default:printf("Error!\n");
}
```

在使用 switch 语句时还应注意以下几点:

(1) <表达式>可以是任何类型,经常采用整型或字符型表达式。

(2) case 后面必须是常量表达式,不能包含变量,且各常量表达式的值不能相同,否则会出现互相矛盾的现象。因为 case 后面的常量表达式实际上起语句标号的作用,而程序中不应该同时出现相同标号的语句。

(3) 在 case 后,允许有多个语句,可以不用 {} 括起来。

(4) 各 case 和 default 子句的先后顺序可以变动,而不会影响程序执行结果。

(5) 多个 case 可以共用同一组执行语句,此时只有在包含语句的 case 中有 break 语句,而其他 case 中则不包含 break 语句。

【例 4.7】 输入任一平年的月份,输出该月份对应的天数。

分析:设月份为 month,天数为 day,则有:

month=1,3,5,7,8,10,12 day=31

month=4,6,9,11 day=30

month=2 day=28

程序如下:

```
#include "stdio.h"
void main()
{
    int month,day;
    printf("请输入月份(1-12):");
    scanf("%d",&month);
    switch (month)
    {
    case 4:
```

```
        case 6:
        case 9:
        case 11:day=30;break;
        case 2:day=28;break;
        default:day=31;
    }
    printf("day=%d",day);
}
```

可以看到，当 month 的值为 4, 6, 9, 11 时有相同的天数，共用语句 `day=30;break;`。

(6) switch 语句也可以内嵌在某个 case 语句中使用，还可以和 if 语句或 if 语句的嵌套形式互相嵌套使用。

到目前为止，处理多分支问题常采用 if-else if-else 语句或 switch 语句来实现。那么，这两种语句在使用中有什么区别呢？

比较【例 4.5】和【例 4.7】可知：if-else if-else 语句与 switch 语句是多分支选择的两种形式。它们的应用环境不同：if-else if-else 语句用于对多条件并列测试，从中取一的情形；switch 语句用于单条件测试，从其多种结果中取一的情形。

4.3 程序举例

【例 4.8】 输入三个整数，输出最大数和最小数。

```
#include "stdio.h"
void main()
{
    int a,b,c,max,min;
    printf("请输入 a,b,c 的值\n");
    scanf("%d,%d,%d",&a,&b,&c);
    if (a>b)
        {max=a;min=b;}
    else
        {max=b;min=a;}
    if (max<c)
        max=c;
    else
        if (min>c)
            min=c;
    printf("max=%d\nmin=%d",max,min);
}
```

本程序中，首先比较 a, b 的大小，把大数装入 max，小数装入 min 中。然后再与 c 比较，若 max 小于 c，则把 c 赋予 max；如果 c 小于 min，则把 c 赋予 min。因此 max 总是最大数，而 min 总是最小数。最后输出 max 和 min 的值即可。

【例 4.9】 计算器程序。用户输入运算数和四则运算符，输出计算结果。

```
#include "stdio.h"
void main()
```

```

{
    float a,b;
    char c;
    printf("请输入运算数和四则运算符如 a+(-,*,/)b:");
    scanf("%f%c%f",&a,&c,&b);
    switch (c)
    {
        case '+': printf("%f\n",a+b);break;
        case '-': printf("%f\n",a-b);break;
        case '*': printf("%f\n",a*b);break;
        case '/': printf("%f\n",a/b);break;
        default: printf("Input error!\n");
    }
}

```

本例可用于四则运算求值。switch 语句用于判断运算符，然后输出运算值。当输入运算符不是 +, -, *, / 时，给出错误提示。

【例 4.10】 求一元二次方程 $ax^2+bx+c=0$ 的根。

分析：

一元二次方程的根有下列情况：

当 $a=0, b=0$ 时，方程无解；

当 $a=0, b \neq 0$ 时，方程只有一个实根 $-c/b$ ；

当 $a \neq 0$ 时，方程的根为

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

其中，当 $b^2-4ac \geq 0$ 时有两个实根；当 $b^2-4ac < 0$ 时有两个虚根。

上述描述用图 4-4 所示流程图表示，逻辑关系更为清晰。

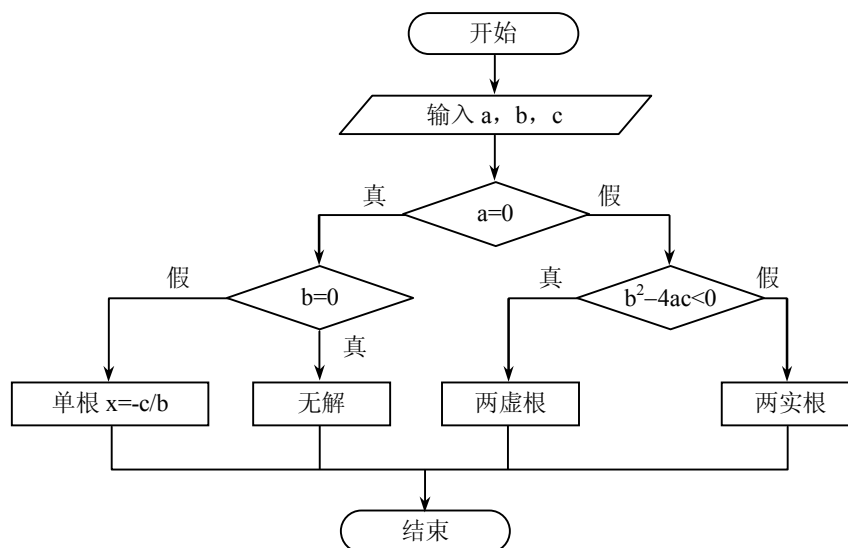


图 4-4 求方程的根

程序如下:

```
#include "stdio.h"
#include "math.h"
void main()
{
    float a,b,c,disc,x1,x2,realpart,imagpart;
    printf("请输入 a,b,c 的值\n");
    scanf("%f,%f,%f",&a,&b,&c);
    if (fabs(a)<=1e-6 && fabs(b)<=1e-6)
        printf("方程无解!\n");
    else if (fabs(a)<=1e-6 && fabs(b)>1e-6)
        printf("方程有单根是%8.4f\n",-c/b);
    else
    {
        disc=b*b-4*a*c;
        if (fabs(disc)<=1e-6||disc>1e-6)
        {
            x1=(-b+sqrt(disc))/(2*a);
            x2=(-b-sqrt(disc))/(2*a);
            printf("方程有两个实根!\n");
            printf("x1=%8.4f\n",x1);
            printf("x2=%8.4f\n",x2);
        }
        else
        {
            realpart=-b/(2*a);
            imagpart=sqrt(-disc)/(2*a);
            printf("方程有两个虚根!\n");
            printf("x1=%8.4f+%8.4fi\n",realpart,imagpart);
            printf("x2=%8.4f-%8.4fi\n",realpart,imagpart);
        }
    }
}
```

程序中用 $disc$ 代表 b^2-4ac , 先计算 $disc$ 的值, 以减少以后的重复计算。同时程序中以 $realpart$ 代表实部, 以 $imagpart$ 代表虚部, 以增加可读性。这里要注意一个问题: 由于 a , b , $disc$ (即 b^2-4ac) 为实数, 而实数在计算和存储时会有一些微小的误差, 因此不能直接进行如下判断, 如 $if(a==0.0)$ 。因为这样可能会出现本来是零的量, 由于上述误差而被判别为不等于零而导致结果错误。所以采取的办法是判别 a 的绝对值 $fabs(a)$ 是否小于一个很小的数 (如 10^{-6}), 如果小于此数, 就认为 $a=0$ 。

习题

一、选择题

1. 两次运行下面的程序, 如果从键盘上分别输入 6 和 4, 则输出结果是 ()。

```
main()
{ int x;
  scanf("%d",&x);
  if (x>5)
    printf("%d",x);
  else
    printf("%d\n",x--);
}
```

- A. 7 和 5 B. 6 和 3 C. 7 和 4 D. 6 和 4
2. 阅读下面的程序，说法正确的是（ ）。

```
main()
{ int x=3,y=0,z=0;
  if (x=y+z) printf("* * * *");
  else printf("# # # #");
}
```

- A. 有语法错误，不能通过编译
B. 输出****
C. 可以通过编译，但是不能通过连接，因而不能运行
D. #####
3. 以下程序的输出结果是（ ）。

```
main()
{ int a= -1,b=1,k;
  if ((++a<0)&& ! (b-- <=0))
    printf("%d %d\n",a,b),
  else
    printf("%d %d\n",b,a),
}
```

- A. -1 1 B. 0 1 C. 1 0 D. 0 0
4. 下列关于 switch 语句和 break 语句的描述中，只有（ ）是正确的。
- A. 在 switch 语句中必须使用 break 语句
B. 在 switch 语句中，可以根据需要使用或不使用 break 语句
C. break 语句只能用于 switch 语句中
D. break 语句是 switch 语句的一部分
5. 对 if 语句中表达式的类型，下面正确的描述是（ ）。
- A. 必须是关系表达式
B. 必须是关系表达式或逻辑表达式
C. 必须是关系表达式或算术表达式
D. 可以是任意表达式
6. 多重 if-else 语句嵌套使用时，寻找与 else 配套的 if 方法是（ ）。
- A. 缩排位置相同的 if B. 其上最近的 if
C. 下面最近的 if D. 同行上的 if
7. 以下错误的 if 语句是（ ）。

- A. if (x>y) z=x;
 B. if (x==y) z=0;
 C. if (x!=y) printf("%d",x) else printf("%d",y);
 D. if (x<y) { x++;y--;}
 8. 执行下面的程序后, 输出为 ()。
- ```
main()
{ int m=20;
 switch(m)
 {
 case 19: m+=1;
 case 20: m+=1;
 case 21: m+=1;
 case 22: m+=1;
 }
 printf("%d\\n",m);
}
```
- A. 20                      B. 21                      C. 22                      D. 23  
 9. 设有说明“int x=2,y=3;”, 则“++x>y--?x:y”的值为 ( )。  
 A. 1                      B. 2                      C. 3                      D. 4  
 10. 若希望当 A 的值为奇数时, 表达式的值为“真”, A 的值为偶数时, 表达式的值为“假”。  
 则以下不能满足要求的表达式是 ( )。  
 A. A%2==1              B. !(A%2==0)              C. !(A%2)              D. A%2

## 二、填空题

- switch 语句的 case 表达式可以是\_\_\_\_\_。
- 用 C 语句描述下列命题: a 小于 b 或小于 c\_\_\_\_\_; a 是奇数\_\_\_\_\_。
- 假设变量 a, b 均为整型, 表达式(a=12,b=13,a<b?1:0)的值是\_\_\_\_\_。
- 当 a, b, c 的值分别为 1, 2, 3 时, 以下程序段执行后 a, b, c 的值分别为\_\_\_\_\_、  
 \_\_\_\_\_、\_\_\_\_\_。

```
if(a<c)
 a=c;
else
 a=b;c=b;b=a;
```

- 下面程序的输出结果是-11, 请填空。

```
main()
{ int x=100,a=200,b=50;
 int v1=25,v2=20;
 if (a<b)
 if (b!=50)
 if (!v1)
 x=11;
 else if (v2)
 x=12;
```

```

x=_____ ;
printf("%d",x);
}

```

### 三、阅读程序，写出执行结果

#### 1. main()

```

{
 int a,b,c;
 a=2;b=3;c=1;
 if (a>b)
 if (a>c)
 printf("%d\n",a);
 else
 printf("%d\n",b);
 printf("end\n");
}

```

#### 2. main ()

```

{ int a=2,b=7,c=5;
 switch (a>0)
 {
 case 1: switch (b<0) { case 1: printf("@"); break ;
 case 2: printf("!"); break ;}
 case 0: switch (c==5) { case 1: printf("*") ; break ;
 case 2: printf("#") ; break ;
 default : printf("#") ; break ; }
 default : printf("&");
 }
 printf("\n");
}

```

### 四、编程题

1. 输入三个整数，要求按大小顺序输出。
2. 有一函数：

$$y = \begin{cases} 5x-2 & (x \leq -1) \\ 2x & (-1 < x < 3) \\ 8-3x & (x \geq 3) \end{cases}$$

编写程序，输入  $x$ ，输出  $y$ 。

3. 设计一个判断输入年份是否为闰年的程序。
4. 由键盘输入三个数，计算以这三个数为边长的三角形的面积。

$$(s = \sqrt{p(p-a)(p-b)(p-c)}, \quad p = (a+b+c)/2)$$

5. 企业发放的奖金根据利润提成。利润（用  $I$  表示）低于或等于 10 万元的，奖金可提 10%；利润高于 10 万元，低于 20 万元（ $100000 < I \leq 200000$ ）时，低于 10 万元的部分按 10% 提成，高于 10 万元的部分可提成 7.5%； $200000 < I \leq 400000$  时，低于 20 万元的部分仍按上述

办法提成(下同), 高于 20 万元的部分按 5%提成;  $400000 < I \leq 600000$  时, 高于 40 万元的部分按 3%提成;  $600000 < I \leq 1000000$  时, 高于 60 万元的部分按 1.5%提成;  $I > 1000000$  时, 超过 100 万元的部分按 1%提成。从键盘输入当月利润  $I$ , 求应发奖金总数。

要求:

(1) 用 if 语句编程序。

(2) 用 switch 语句编程序。

6. 判别键盘输入字符的类别。

分析: 可以根据输入字符的 ASCII 码来判别类型。由 ASCII 码表可知 ASCII 值小于 32 的为控制字符。在 0~9 之间的为数字, 在 A~Z 之间为大写字母, 在 a~z 之间为小写字母, 其余则为其他字符。

7. 给一个不多于 5 位的正整数, 要求: ①求出它是几位数; ②分别打印出每一位数字; ③按逆序打印出各位数字。例如原数为 321, 应输出 123。

8. 输入一个 5 位数, 判断它是不是回文数。例如, 12321 是回文数, 个位与万位相同, 十位与千位相同。