# 第5章 分支结构

**学习目标**

- 正确地判断和使用逻辑表达式和关系表达式；
- 掌握各种形式 if 语句的语法和用法，注意嵌套 if 结构中 if 和 else 的匹配关系；
- 掌握 switch 语句语法和使用方法，注意 switch 语句的控制流程；
- 掌握 switch 语句中 break 语句的使用以及 switch 语句的嵌套；
- 熟练使用 if 结构和 switch 结构解决简单的应用问题。

## 5.1 if 结构

在日常生活中大家可能经常遇到这样的问题，如果今天天气好我就出去郊游，否则就在家看电视。对于这种需要根据不同情况执行不同操作的问题，计算机该如何处理？顺序语句只能按照语句的先后顺序来运行，显然不能根据判断结果选择处理方式，这要求程序本身具有判断能力。本章介绍的分支结构正是为解决这一类问题而设定的。本章将会介绍 C 语言中的几种分支结构，包括 if 结构和 switch 结构。

if 结构是最常见的分支结构，if 语句是单分支结构，if-else 构成双分支结构，同时可以利用 if 语句或者 if-else 语句的嵌套实现多分支结构。

### 5.1.1 if 语句

if 语句是最简单的分支语句，是一种单分支结构，其语法形式为：

```
if(<表达式>)
{
    <语句 A>
}
```

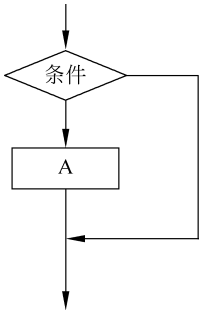


图 5-1 if 结构的流程图

上述结构中的<表达式>一般为条件表达式或者逻辑表达式，用一对小括号括起来。if 语句的功能是先判断<表达式>的逻辑值，若该逻辑值是“真”，则进入分支运行<语句 A>，否则什么也不运行。式中，<语句 A>是分支语句，可以是一条语句也可以是多条语句，如果分支结构只有一条语句。则大括号可以省略。

if 结构的流程图如图 5-1 所示。

**【例 5-1】** 若有定义 int a＝3，b＝1；，判断下列表达式逻辑值

真假。

a＞0　　a＞＝0　　a＞b　　a＝＝b　　a!＝b　　a＝＝1　　a＝1

a＝b　　a＝0　　a&&b　a||b　　！a　　!b－－　　!－－b

分析：a＞0 为真，a＞＝0 为真，a＞b 为真，a＝＝b 为假，a!＝b 为真，a＝＝1 为假，a＝1 为真，a＝b 为真，a＝0 为假，a&&b 为真，a||b 为真，!a 为假，!b－－为假，!－－b 为真。

**注意**：一个等号表示赋值，两个等号表示比较大小。a＝b，则 a 的值由 3 变为 1；a＝＝b 是判断 a 和 b 的值是否相等，相等则结果为 1，不等则结果为 0。

**【例 5-2】** 从键盘任意输入一个实数 a，求其绝对值并输出。

分析：根据题目要求，当 a 是负数时，需要求其相反数；当 a 为非负数，则不需要做任何改变。

```
1   #include<stdio.h>                          //包含头文件 stdio.h,因为用到输入输出函数
2   void main(){                               //主函数名字固定为 main
3       float a;                               //定义单精度浮点型变量,可以存小数
4       printf("请输入一个实数: ");             //在屏幕上输出提示
5       scanf("%f", &a);                       //从键盘输入一个浮点数,存到变量 a 中
6       if (a < 0)                             //判断输入的数小于 0
7       {
8           a = -a;                            //负数变为正数
9       }
10     printf("a 的绝对值为: %f\n", a);        //输出绝对值数值
11  }
```

运行结果：

```
请输入一个实数：-10
a的绝对值为：10.000000
```

```
请输入一个实数：7.7
a的绝对值为：7.700000
```

**第三次提示**：如果使用 VC 2010 环境，可在 main 函数体语句最后面，结尾大括号前面，增加 getchar();语句，避免画面一闪而过。

## 5.1.2 if-else 语句

if-else 语句是由关键字 if 和 else 构成的二分支结构，其语法形式为：

```
if(<表达式>)
{
    <语句 A>
} else
{
    <语句 B>
}
```

上述结构中的<表达式>一般为条件表达式或者逻辑表达式，用一对小括号括起来。if-else 语句的功能是先判断<表达式>的逻辑值，若该逻辑值是“真”，则运行<语句 A>，否则运行<语句 B>。式中，<语句 A>、<语句 B>都是分支语句，可以是一条语句也可以是多条语句，如果只有一条语句，则大括号可以省略。一次条件判断结束，<语句 A>、<语句 B>只可能运行其一，不可能同时都运行。

if-else 结构的流程图如图 5-2 所示。

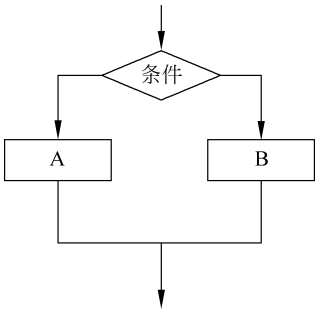


图 5-2　if-else 结构流程图

**【例 5-3】** 从键盘输入一个整数 n,判断该数的奇偶,如果 n 是偶数输出“n 是偶数”;如果 n 不是偶数输出“n 是奇数”。

分析:根据题目要求,判断 n 是否为偶数,也就是 n 除以 2 的余数是否为 0。根据不同的结果需要输出不同的信息,很显然应该使用二分支结构。

```
#include<stdio.h>                          //包含头文件 stdio.h,因为用到输入输出函数
void main(){                               //主函数名字固定为 main,返回值为空值
  int n;                                   //定义变量
  printf("请输入一个整数:");                //在屏幕上输出提示
  scanf("%d", &n);                         //从键盘输入一个整数,存入变量 n 中
  if (n%2==0)                              //余数等于 0 则为真,不等于 0 则为假
  {  printf("%d是偶数\n", n); }
  else
  {  printf("%d是奇数\n", n); }
}
```

运行结果:

```
请输入一个整数:12
12是偶数
```

```
请输入一个整数:15
15是奇数
```

上述程序也可以写成这样,观察一下判断条件的变化。

```
#include<stdio.h>
void main(){
    int n;
    printf("请输入一个整数:");
    scanf("%d", &n);
    if (n%2)                               //余数为 1 则为真,余数为 0 则为假
    {  printf("%d是奇数\n", n);  }
    else
    {  printf("%d是偶数\n", n);  }
}
```

**【例 5-4】** 编写程序,通过输入 $x$ 的值,计算分段函数 $y$ 的值。

$$y=\begin{cases}x, & x\geqslant 0\\ x^2, & x<0\end{cases}$$

分析：根据题目要求，对于输入的 $x$ 值要进行判断，在 $x<0$ 和 $x\geqslant 0$ 的情况下，$y$ 的取值不同。

```
#include<stdio.h>
void main(){
    float x, y;
    printf("请输入一个实数: ");
    scanf("%f", &x);                    //读入一个单精度浮点数
    if (x<0)
    {  y=x*x;  }
    else
    {  y=x;  }
   printf("y=%f\n", y);
}
```

运行结果：

```
请输入一个实数：-6
y=36.000000
```

```
请输入一个实数：6
y=6.000000
```

### 5.1.3 if 语句的嵌套

if 语句的嵌套是指在 if 或者 else 的分支结构中包含另外的 if 语句或者 if-else 语句，也就是前面所述的 if 结构中<语句 A>、<语句 B>可以是 if 语句或者 if-else 语句。if 语句的嵌套层次原则上可以是任意多层，但通常情况下不建议使用层数较多的嵌套。if 语句的嵌套形式一般分为两种：规则嵌套和不规则嵌套。

**1. 规则嵌套**

规则嵌套是每一层的 if 或 else 分支嵌套着另一个 if-else 结构。其语法形式如下：

```
if(<表达式 1>)
{
    <语句 A>
}
else if(<表达式 2>)
{
    <语句 B>
}
else if(<表达式 3>)
{
    <语句 C>
}
…
else if(<表达式 n>)
{
    <语句 X>
}
else
{
    <语句 X+1>
}
```

上述结构的功能是先判断<表达式 1 >的逻辑值，若该逻辑值是“真”，则运行<语句 A >，否则判断<表达式 2 >的逻辑值，若为“真”，则运行<语句 B >，否则继续判断<表达式 3 >的逻辑值……以此类推。

规则嵌套的流程图如图 5-3 所示。

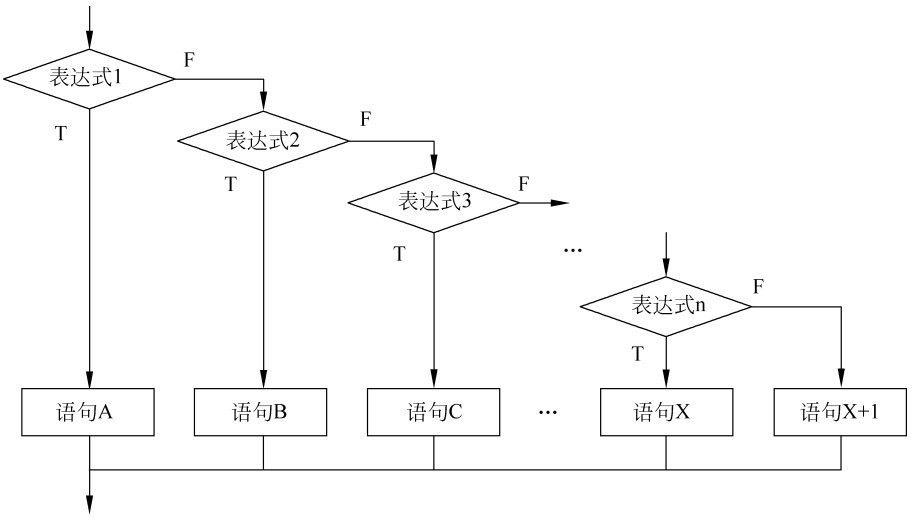


图 5-3　if-else 的规则嵌套流程图

**【例 5-5】** 编写程序，通过输入 $x$ 的值，计算分段函数 $y$ 的值。

$$y=\begin{cases}x & (x<1)\\ 2x-1 & (1\leqslant x<10)\\ 3x-1 & (x\geqslant 10)\end{cases}$$

分析：根据题目要求，对于输入的 $x$ 值要进行判断。在 $x<1$、$1\leqslant x<10$ 和 $x\geqslant 10$ 三种情况下，$y$ 值的计算公式不同。

```
1   #include<stdio.h>
2   void main()
3   {
4       float x, y;
5       printf("请输入一个实数: ");
6       scanf("%f",&x);
7       if(x<1)
8       { y=x;}
9       else if(x<10)
10     {   y=2*x-1;   }
11     else
12     {   y=3*x-1;   }
13     printf("y的值为: %f\n", y);
14  }
```

运行结果：

```
请输入一个实数：0.45
y的值为：0.450000
```

```
请输入一个实数：5.4
y的值为：9.800000
```

```
请输入一个实数：10.1
y的值为：29.300001
```

**问题**：else if 语句中为何不需要写为 x>=1 && x<10，却只写 x<10 条件？

**【例 5-6】** 从键盘输入学生百分制成绩 score，输出成绩等级 A、B、C、D。A 等为 85 分以上，B 等为 70～84 分，C 等为 60～69 分，D 等为 60 分以下。

分析：将百分制成绩分为 4 个等级输出，需要进行多层判断。

```
#include<stdio.h>
void main()
{
    int score;
    printf("请输入一个百分制成绩: ");
    scanf("%d",&score);
    if(score>=0&&score<=100)
    {
      if(score>=85)
     {
        printf("成绩为 A 等\n");
     }else if(score>=70)
     {
        printf("成绩为 B 等\n");
     }else if(score>=60)
     {
        printf("成绩为 C 等\n");
     }else
     {
        printf("成绩为 D 等\n");
     }
  }else
     printf("您输入的不是百分制分数\n");
}
```

运行结果：

```
请输入一个百分制成绩：90
成绩为A等
请输入一个百分制成绩：80
成绩为B等
请输入一个百分制成绩：-10
您输入的不是百分制分数
```

在使用 if-else 嵌套结构时需要注意以下几点。

(1) C 语言规定，else 总是与在它上面、距它最近且尚未匹配的 if 配对。

(2) 程序书写采用缩进方式，将同一层的分支结构对齐，这样可以增加程序的美观性和可读性。

(3) 程序设计时，if-else 结构不宜太多，2～3 层左右，否则对程序的运行效率会有所影响。

**2. 任意嵌套**

与规则嵌套不同，任意嵌套是在 if-else 结构中的任一分支中嵌入 if 结构或者 if-else 结构。实际问题中，大部分都是任意嵌套的形式。

对于初学者来说，多重 if-else 结构最容出错的就是 if 与 else 的配对问题。建议大家在编写程序时，先写好分支结构的左右大括号，再在大括号中间编写分支语句或者其他的分支结构，避免遗漏右边大括号造成的程序错误。同时建议，对于只有一条语句的分支也不要省

略大括号,避免出现不必要的逻辑错误。

【例 5-7】 试比较下列两段程序的差异。

程序一:



```
#include<stdio.h>
void main()
{
    int a=1,b=-1;
    if(a>0)
        if(b>0)
            a++;
        else a--;
    printf("a=%d\n",a);
}
```

运行结果:

```
a=0
```

程序二:

```
#include<stdio.h>
void main()
{
    int a=1,b=-1;
    if(a>0)
    {
        if(b>0)
            a++;
    }
    else a--;
    printf("a=%d\n",a);
}
```

运行结果:

```
a=1
```

## 5.2 switch 结构

通过上面的学习我们了解到,对于多种分支的情况,我们可以采用 if 语句和 if-else 语句的多重嵌套来实现。但是如果需要判断的分支很多,嵌套层次就会变得复杂,而且程序运行效率会降低。为了解决这个问题,C 语言提供了 switch 语句专门处理多路分支的情形。

switch 语句是一种多路分支语句,其语法形式如下:

```
switch(<表达式>)
{
case <值 1>: 语句序列 1;
             break;
case <值 2>: 语句序列 2;
             break;
```

```
    …
    case <值 n-1>: 语句序列 n-1;
                break;
    case <值 n>: 语句序列 n;
           break;
    default: 语句序列 n+1;
}
```

其运行过程为：当 switch 后面的<表达式>的值与某个 case 后面的“值 i”相同时，就运行 case 后面的语句序列 i；当运行到 break 语句时，跳出 switch 语句，运行后面的程序语句。如果没有任何一个 case 后面的值与<表达式>的值相等，则运行 default 后面的语句序列，直到 switch 语句结束。

switch 结构可以没有 default 标号，此时如果没有任何一个 case 后面的值与<表达式>的值相等，则直接跳出 switch 结构。

switch 语句运行的流程图如图 5-4 所示。

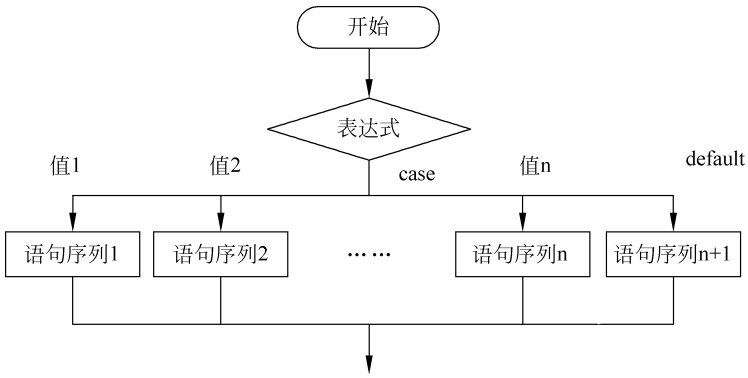


图 5-4　switch 语句流程图

**【例 5-8】** 输入成绩等级 A、B、C、D，要求按照考试成绩的等级输出百分制分数段。A 等为 85 分以上，B 等为 70～84 分，C 等为 60～69 分，D 等为 60 分以下。

分析：这是一个多分支选择问题，根据 4 个等级输出对应的百分制分数范围。如果用 if 语句来至少要三层的嵌套，使用 switch 则只需一次匹配即可。

```
1   #include<stdio.h>
2   void main(){
3       char grade;
4       printf("请输入一个成绩等级: ");
5       scanf("%c",&grade);                    //读入一个字符
6       switch(grade)                          //switch 多分支
7       {
8          case 'A': printf("85-100\n");break;
9          case 'B': printf("70-84\n");break;
10         case 'C': printf("60-69\n");break;
11         case 'D': printf("60 以下\n");break;
12         default:printf("输入错误!\n");
13     }
14  }
```

运行结果：

```
请输入一个成绩等级：A
85-100
请输入一个成绩等级：D
60以下
请输入一个成绩等级：F
输入错误！
```



在使用 switch 语句时还应注意以下几点。

(1) switch 后面的<表达式>，一般是 int 型、char 型和枚举型中的一种，不能是 float 型或者 double 型。同时，每个 case 后面“值”的类型应该与 switch 后面的<表达式>的类型一致。

(2) case 后面的语句序列可以是一条语句也可以是多条语句，但语句序列的大括号可以省略，如：

```
switch(i){
    case 1:  { a++;  b++;  break;}          //  {  }可省略
    case 2:  a--;  b--;  break;             //省略了{  }
    default:  break;
}
```

(3) 每个 case 后面的数值必须各不相同，否则会出现相互矛盾的现象。因为对<表达式>的同一值，不允许有两种或以上的运行方案。

(4) 每个 case 后面的数值必须是常量或者是常量表达式，不能包含变量。

(5) case 后面的数值仅起语句标号作用，并不进行条件判断。系统一旦找到入口标号，就从此标号开始运行，不再进行标号判断，直到遇见 break 语句跳出，或者运行到 switch 语句结束。例如，在例 5-8 中，如果去掉各个 case 子句中的 break 语句，将连续输出，效果如下。

```
1   #include<stdio.h>
2   void main(){
3       char grade;
4       printf("请输入一个成绩等级: ");
5       scanf("%c",&grade);
6       switch(grade)
7       {
8          case 'A':  printf("85-100\n");
9          case 'B':  printf("70-84\n");
10         case 'C':  printf("60-69\n");
11         case 'D':  printf("60以下\n");
12         default:  printf("输入错误!\n");
13     }
14  }
```

运行结果：

```
请输入一个成绩等级：A
85-100
70-84
60-69
60以下
输入错误！
```

**提示**：如果 case 子句和 default 子句都带有 break 语句，那么它们之间的顺序不影响

switch 分支的功能。如果 case 子句和 default 子句有的带有 break 语句，有的没有，那么它们之间的顺序变化可能会影响输出结果。

## 5.3 程序范例

**【例 5-9】** 已知某公司员工的保底月薪为 1500，某月所接工程的绩效 profit(整数)与绩效提成的关系如下所示，计算员工的当月薪水。

| 工程绩效 profit | 提成比率 |
|---|---|
| profit < 1000 | 没有提成 |
| 1000 <= profit < 2000 | 10% |
| 2000 <= profit < 5000 | 15% |
| 5000 <= profit < 10000 | 20% |
| profit >= 10000 | 25% |

分析：这是一个典型的多分支问题，员工最终的薪水＝保底月薪＋工程绩效 * 提成比率，可以使用 if-else 的嵌套结构，也可以使用 switch 结构。

方法一：

```
1   #include<stdio.h>                          //包含头文件
2   void main()                                //主函数
3   {
4       int profit;                            //定义变量
5       float salary=0;
6       printf("请输入当月绩效: ");
7       scanf("%d",&profit);                   //输入绩效
8       if(profit<1000)                        //1000 以下没有提成
9       {
10          salary = 1500;
11      }else if(profit<2000)                  //大于等于 1000,小于 2000,提成 10%
12      {
13          salary = 1500+profit*0.1;
14      }else if(profit<5000)                  //大于等于 2000,小于 5000,提成 15%
15      {
16          salary = 1500+profit*0.15;
17      }else if(profit<10000)                 //大于等于 5000,小于 10000,提成 20%
18      {
19          salary = 1500+profit*0.2;
20      }else                                  //大于等于 10000,提成 25%
21      {
22          salary = 1500+profit*0.25;
23      }
24      printf("当月薪水为: %.2f",salary);     //输出薪水
25  }
```

运行结果：

```
请输入当月绩效：800
当月薪水为：1500.00
```

```
请输入当月绩效：8000
当月薪水为：3100.00
```

方法二：

使用 switch 语句，需要对 profit 值判断，但是 profit 可能是很大的值，不可能在程序中一一穷举。分析本题可知，提成的变化都是 1000 的整数倍，我们可以将 profit 值除以 1000 再进行匹配。

```
#include <stdio.h>
void main()
{
    int profit;
    float salary = 0;
    printf("请输入当月绩效: ");
    scanf("%d",&profit);
    switch(profit/1000)                          //判断绩效除以 1000 的值
    {
    case 0: salary = 1500; break;                //1000 以下没有提成
    case 1: salary = 1500 + profit * 0.1;break;  //1000 = <绩效< 2000,提成 10 %
    case 2:
    case 3:
    case 4: salary = 1500 + profit * 0.15;break; //2000 = <绩效< 3000,提成 15 %
    case 5:
    case 6:
    case 7:
    case 8:
    case 9: salary = 1500 + profit * 0.2;break;  //5000 = <绩效< 10000,提成 20 %
    default: salary = 1500 + profit * 0.25;      //绩效大于 10000,提成 25 %
    }
    printf("当月薪水为: %.2f",salary);
}
```

运行结果：

```
请输入当月绩效：800
当月薪水为：1500.00
```

```
请输入当月绩效：8000
当月薪水为：3100.00
```

**提示**：不是所有的多分支都能用 switch 语句，能枚举每一个可能取值才能用 switch 语句，但所有的多分支都可以用嵌套 if-else 语句实现。

**【例 5-10】** 写一程序，从键盘输入年份 year，判断其是否为闰年。闰年的条件是：能被 4 整除但不能被 100 整除，或者能被 400 整除。

分析：如果 X 能被 Y 整除，则余数为 0，即 X%Y==0。

```
#include <stdio.h>
void main()
{
    int year;
    printf("请输入年份: ");
    scanf("%d", &year);
    if(year % 4 == 0 && year % 100 != 0)         //能被 4 整除,不能被 100 整数
    {
        printf("%d 是闰年\n", year);
    }else if(year % 400 == 0)                    //能被 400 整除
```

```
11      {
12          printf("%d是闰年\n", year);
13      }else
14      {
15          printf("%d不是闰年\n", year);
16      }
17  }
```

运行结果：

```
请输入年份：2000
2000是闰年
请输入年份：2012
2012是闰年
请输入年份：2019
2019不是闰年
```

## 本章小结

程序设计三种基本结构是：顺序结构、分支结构、循环结构。C语言中分支结构通过if-else语句和switch语句实现。

if-else语句可以实现单分支和双分支结构，使用嵌套if-else结构可以实现多分支。

switch语句可以实现多分支结构，但不是所有的多分支都可以用switch结构实现。switch结构中要注意用好break语句。

if-else语句和switch语句可以嵌套使用。

## 习　　题

### 一、选择题

1. 以下程序的运行结果是(　　)。

```
#include <stdio.h>
int main()
    { int m=5;
      if (m++>5) printf ("%d\n", m);
      else       printf ("%d\n", m--);  }
```

A. 4　　B. 5　　C. 6　　D. 7

2. 以下程序的运行结果为(　　)。

```
int a=1, b=2, c=2, t=0;
if(a<b) {t=a;a=b;b=t;c++;}
printf("%d, %d, %d",a, b, c);
```

A. 1，2，0　　B. 2，1，0

C. 1，2，1　　D. 2，1，3

3. 设有int a=1，b=2，c=3，d=4，m=2，n=2;,运行(m=a>b)&&(n=c>d)后n的值是(　　)。

A. 1　　B. 2

C. 3　　D. 4

4. 对 if 语句中表达式的类型,下面说法正确的是(　　)。

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

5. 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--;}

6. 已知 int x=10, y=20, z=30;,语句 if(x>y) z=x;x=y;y=z;运行后,x, y, z 的值是(　　)。

A. x=10, y=20, z=30　　B. x=20, y=30, z=30

C. x=20, y=30, z=10　　D. x=20, y=30, z=20

7. 当 a=1, b=3, c=5, d=4 时,运行完下面一段程序后 x 的值是(　　)。

```
if(a<b)
  if(c<d)  x=1;
  else
    if(a<c)
      if(b<d)  x=2;
      else  x=3;
    else  x=6;
else  x=7;
```

A. 1　　B. 2　　C. 3　　D. 6

8. 有如下程序,正确的输出结果是(　　)。

```
#include<stdio>
void main()
{  int a=15, b=21, m=0;
   switch(a%3)
   { case 0:m++;break;
     case 1:m++;
     switch(b%2)
       {  default:m++;
          case 0:m++;break;  }
   }
   printf("%d\n", m);
}
```

A. 1　　B. 2　　C. 3　　D. 4

9. 若 a, b, c1, c2, x, y 均是整型变量,错误的 switch 语句是(　　)。

A.
```
switch(a+b)
  {
    case 1: y=a+b; break;
    case 0: y=a-b; break;
    case 3: y=b-a; break;
  }
```

B.
```
switch(a*a+b*b)
    {
      case 3:break;
      case 1: y=a+b; break;
    }
```

C. switch a
{
case c1: y=a+b; break;
case c2: y=a-b; break;
default: y=b-a;
}

D. switch(a-b)
{
default: y=a*b;break;
case 3:
case 4: y=b-a; break;
}

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

11. 若有定义语句 int a, b; double x;，则下列选项中没有错误的是(　　)。

A. switch (x%2)
{ case 0: a++; break;
case 1: b++; break;
default: a++; b++; }

B. switch ((int)x/2.0)
{ case 0: a++; break;
case 1: b++; break;
default : a++; b++; }

C. switch((int)x%2)
{ case 0: a++; break;
case 1: b++; break;
default: a++; b++; }

D. switch((int)(x)%2)
{ case 0.0: a++; break;
case 1.0: b++; break;
default : a++; b++; }

12. 以下选项中与 if(a==1) a=b;else a++;语句功能不同的 switch 语句是(　　)。

A. switch(a) {case 1:a=b;break; default:a++; }

B. switch(a==1) {case 0:a=b;break; case 1:a++; }

C. switch(a) {default:a++;break; case 1:a=b; }

D. switch(a==1) {case 1:a=b;break; case 0:a++; }

13. 选项中与如下嵌套 if 语句等价的语句是(　　)。

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

A. k=(a<b)? a:b; k=(b<c)? b:c;

B. k=(a<b)? (b<c? a:b):(b>c)? b:c);

C. k=(a<b)? (a<c? a:c):(b<c? b:c);

D. k=(a<b)? a:b; k=(a<c)? a:c;

14. 以下程序的运行结果是(　　)。

```
#include<stdio.h>
int main()
{ int x=1, y=0;
  if(!x) y++;
  else if(x==0)
     if (x)  y+=2;
     else    y+=3;
```

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

A. 3　　B. 2　　C. 1　　D. 0

15. 以下程序的运行结果是(　　)。

```
#include <stdio.h>
int main()
{ int x = 1, y = 2, z = 3;
  if(x > 1)
  if(y > x) putchar('A');
   else    putchar('B');
  else
    if(z < x) putchar('C');
    else     putchar('D');  }
```

A. A　　B. B　　C. C　　D. D

16. 以下叙述中正确的是(　　)。

A. if 语句只能嵌套一层

B. 改变 if-else 语句的缩进格式,会改变程序的运行流程

C. 不能在 else 子句中再嵌套 if 语句

D. if 子句和 else 子句中可以是任意的合法的 C 语句

17. 以下程序的运行结果是(　　)。

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

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

18. 以下程序的运行结果是(　　)。

```
#include <stdio.h>
void main()
{    float x = 2.0, y;
     if(x < 0.0)   y = 0.0;
     else if(x < 10.0)   y = 1.0/x;
     else   y = 1.0;
     printf("%f\n", y);    }
```

A. 0.000000　　B. 0.250000　　C. 0.500000　　D. 1.000000

## 二、填空题

1. C 语言用(　　)表示假,(　　)表示真。

2. 写出下列各逻辑表达式的值。设 a=3,b=4,c=5。

(1) a+b > c&&b==c　　(　　)

(2) a||b+c&&b-c　　(　　)

(3) !(a>b)&&! c||1　　(　　)

(4) !(x==a)&&(y==b)&&0　　(　　)

(5) !(a+b)+c-1&&b+c/2　　(　　)

3. 以下程序的运行结果是(　　)。

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

4. 以下程序的运行结果是(　　)。

```
#include <stdio.h>
int main()
{
  int ch1 = 0, ch2 = 5;
  if(ch1!= 3) printf("ch1: %d", ch1);
  else        printf("ch2: %d", ch2);
}
```

5. 若从键盘输入 58,以下程序的运行结果是(　　)。

```
#include <stdio.h>
int main()
{
  int a;
  scanf("%d", &a);
  if(a > 50)    printf("%d", a);
  if(a > 40)    printf("%d", a);
  if(a > 30)    printf("%d", a);
}
```

6. 以下程序的运行结果是(　　)。

```
#include <stdio.h>
int main()
{
  int a = 100;
  if(a > 100)    printf("%d\n", a > 100);
  else           printf("%d\n", a <= 100);
}
```

7. 以下程序的运行结果是(　　)。

```
#include <stdio.h>
int main()
{ int a = 1, b = 0;
  switch(a)
```

```
    {  case 1: switch (b)
               {  case 0: printf("**0**"); break;
                  case 1: printf("**1**"); break;}
       case 2: printf("**2**"); break;
    }
}
```

**三、编程题**

1. 编写程序，输入一个整数，判断该数的正负性和奇偶性。

2. 从键盘输入一个字符，如果是大写字母，就转换成小写；如果是小写字母，就转换成大写；如果是其他字符原样保持并将结果输出。

3. 从键盘输入一个数，判断其是否是5的倍数而不是7的倍数。如果是，输出Yes，否则输出No。

4. 编写程序，输入一个整数，当其为65时输出A，66时输出B，67时输出C，其他值时输出END。