

第4章

选择结构程序设计

本章学习目标

本章主要介绍 C 语言的 if 语句、switch 语句的格式和使用方法及选择结构程序设计方法等内容。通过本章的学习，要求达到以下目的：

- 熟悉 if 语句的格式及应用的几种形式
- 掌握 if 语句嵌套的应用方法
- 掌握条件运算符的应用
- 掌握 switch 语句的格式及应用
- 掌握选择结构程序的设计方法

在程序设计中，常常需要根据给定的条件决定执行相应的不同操作。在多条可执行路径中，根据条件选择且只能选择其中一条路径执行的程序结构称为选择结构。选择结构是结构化程序设计的三种基本结构之一。

在 C 语言中，一般用关系表达式或逻辑表达式表示选择结构的条件，用 if 语句或 switch 语句实现选择结构。

4.1 if 语句

1. if 语句的一般格式

if(表达式) {语句组 1}

[else {语句组 2}]

如：if(a>=b) printf("%d", a);

else printf("%d", b);

说明：

(1) if 语句中的表达式必须用一对圆括号“()”括起来。

(2) else 子句（可缺省）是 if 语句的一部分，必须与 if 配对使用，不能单独使用。

(3) 当 if 和 else 下面的语句组，仅由一条语句构成时，可以不使用复合语句形式（即不必用花括号“{}”括起来）。

2. if 语句的三种形式

形式一：缺省 else 子句的 if 语句。

格式：**if (表达式) 语句**

当“表达式”的值不等于 0（即判定为“逻辑真”）时，则执行语句，否则直接转向执行下一条语句，如图 4-1 所示。

例如：if(x>y) printf("max=%d",x); /*输出 x>y，则输出 x 的值，否则不输出*/

形式二：含有 else 子句的 if 语句。

格式：if(表达式)语句 1 else 语句 2

当“表达式”的值不等于 0（即判定为“逻辑真”）时，则执行语句 1；否则，执行语句 2。流程图如图 4-2 所示。

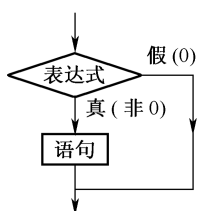


图 4-1 缺省 else 子句的 if 语句流程图

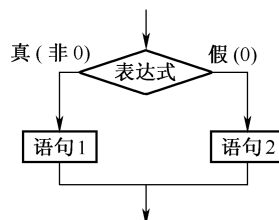


图 4-2 if-else 语句的流程图

例如：if(x>y) printf("max=%d",x);

else printf("max=%d",y);

该语句的功能为输出 x、y 中的较大的值。

形式三：嵌套的 if 语句。

格式：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 n

以上是 if 语句中又包含 if 语句，形成 if 语句嵌套的情形，其功能是：先判断“表达式 1”的值，若值为真，则执行“语句 1”；否则判断“表达式 2”的值，若值为真，则执行“语句 2”；否则继续判断。当出现某个表达式的值为真时，执行相应的语句。然后跳到整个 if 语句之外继续执行程序。如果所有的表达式均为假，则执行“语句 n”，如图 4-3 所示。

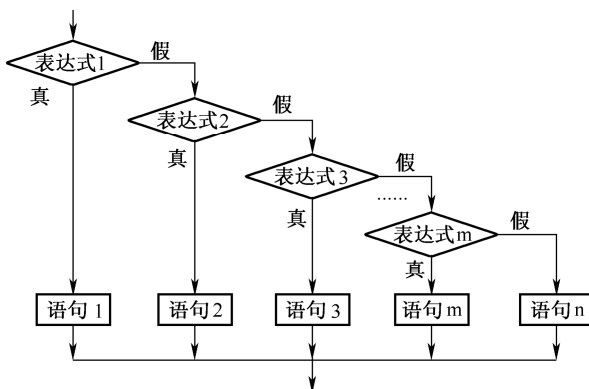


图 4-3 if-else 语句嵌套的流程图

```
例如: if(number>500) const=0.15;
      else if(number>300) const=0.10;
        else if(number>100) const=0.075;
          else if(number>50) const=0.05;
            else const=0;
```

说明:

(1) if(表达式)中的“表达式”一般为逻辑表达式或关系表达式, 如:

```
if(a==b && x==y) printf("a=b,x=y");
```

也允许是其他类型的数据, 如整型、实型、字符型等。例如:

```
if(3) printf("O.K.");
```

```
if(3.5) printf("%f", 3.5);
```

```
if('a') printf("%d", 'a');
```

(2) “语句组 1”和“语句组 2”可以只包含一个简单语句, 也可以是复合语句。复合语句须用一对花括号括起来, 例如:

```
if(a+b>c && b+c>a && c+a>b)
{
    s=0.5*(a+b+c);
    area=sqrt(s*(s-a)*(s-b)*(s-c));
    printf("area=%8.2f",area);
}
else
printf("所输入的三边长不能构成三角形");
```

(3) 不管是简单语句, 还是复合语句中的各个语句, 每个语句后面的分号必不可少! 例如:

```
if(x>0)
    printf(" %f", x);
else
    printf(" %f", -x);
```

if 行后面的“printf(" %f", x);”语句中分号不能省略, 同理 else 行后面的“printf(" %f", -x);”语句中分号也不能省略。

例 4-1 输入任意三个整数, 求三个数中的最大值。

提示: 定义三个变量 num1、num2、num3, 分别表示从键盘输入的任意三个数, 再定义变量 max 表示三个数中的最大值。

首先取一个数预置为 max (最大值), 然后再用 max 依次与其余的数逐个比较, 如果发现比 max 大的, 就用它给 max 重新赋值, 比较完所有的数后, max 中的数就是最大值。

对于从 3 个或 3 个以上的数中找最大值的处理, 上述方法是非常好用的。

程序如下:

```
/*exam4-1*/
main(void)
{
    int num1,num2,num3,max;
    printf("Please input three numbers:");
    scanf("%d,%d,%d",&num1,&num2,&num3);
    max=num1;
    if(num2>max) max=num2;
```

```

    if(num3>max)
        max=num3;
    printf("The three numbers are:%d,%d,%d\n",num1,num2,num3);
    printf("max=%d\n",max);
}

```

程序运行情况如下：

```
Please input three numbers:35,100,85✓
```

```
The three numbers are: 35,100,85
```

```
max=100
```

例 4-2 输入任意三个数 num1、num2、num3，按从小到大的顺序排序输出。

提示：编程思路与例 4-1 基本上相同。不同的是：由于本例题要求将所输入的三个数按从小到大的顺序排序输出，故在比较大小时要注意小数排在前，大数排在后，位置不符时要交换两数的位置。

程序如下：

```

/*exam4-2*/
main(void)
{
    int num1,num2,num3,temp;
    printf("Please input three numbers:");
    scanf("%d,%d,%d",&num1,&num2,&num3);
    if(num1>num2) {temp=num1;num1=num2;num2=temp;}
    if(num2>num3) {temp=num2;num2=num3;num3=temp;}
    if(num1>num2) {temp=num1;num1=num2;num2=temp;}
    printf("Three numbers after sorted: %d,%d,%d\n",num1,num2,num3);
}

```

程序运行情况如下：

```
Please input three numbers: 35,100,85✓
```

```
Three numbers after sorted: 35, 85,100
```

3. if 语句的嵌套

if 语句允许嵌套。所谓 if 语句的嵌套是指：if 语句中包含另一个 if 语句，即在“语句组 1”或（和）“语句组 2”中，又包含有 if 语句的情况。

（1）if 语句的嵌套形式

if 语句的嵌套的一般形式：

if(表达式 1)

if(表达式 2) 语句组 1 /*内嵌 if 语句*/

else 语句组 2

else

if(表达式 3) 语句组 3 /*内嵌 if 语句*/

else 语句组 4

（2）if 语句的嵌套与嵌套匹配原则。

if 语句嵌套时，从最内层开始，else 总是与它上面距它最近且尚未匹配的 if 配对。

为明确匹配关系，避免 if 与 else 配对错位的最佳办法是将内嵌的 if 语句，一律用花括号括起来。

另外，可使用适当的缩进，保证 if 和 else 不错位配对，而且便于阅读程序。例如：

if()		if()
if() 语句 1		{if() 语句 1}
else	加 {} 改变配对关系：	else
语句 2		语句 2

例 4-3 有一函数如下，

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

请编程实现：输入一个 x 值，即可输出 y 值。

下面以不同的 if 语句嵌套的方法来实现。

方法一：

```
/*exam4-3-1*/
main(void)
{
    int x,y;
    scanf("%d",&x);
    if(x<0) y=-1;
    else if(x==0) y=0;
    else y=1;
    printf("x=%d,y=%d\n",x,y);
}
```

当任意输入 x 的值为 35 时，屏幕显示程序的运行情况如下：

35 ✓

x=35,y=1

方法二：

```
/*exam4-3-2*/
main(void)
{
    int x,y;
    scanf("%d",&x);
    if(x>=0)
        if(x>0) y=1;
        else y=0;
    else y=-1;
    printf("x=%d,y=%d\n",x,y);
}
```

方法三：

```
/*exam4-3-3*/
main(void)
{
    int x,y;
    scanf("%d",&x);
    y=-1;
    if(x>=0)
```

```
if(x>0) y=1;
else y=0;
printf("x=%d,y=%d\n",x,y);
}
```

方法四:

```
/*exam4-3-4*/
main(void)
{
    int x,y;
    scanf("%d",&x);
    y=0;
    if(x!=0)
        if(x>0) y=1;
        else y=-1;
    printf("x=%d,y=%d\n",x,y);
}
```

4. 条件运算符——“?”运算符

在 if 语句中,当“表达式”的值为“真”和“假”时,都只执行一个赋值语句给同一个变量赋值,例如:

```
if(a>b) max=a;
else max=b;
```

则可以用条件运算符构成一个条件表达式来处理,例如上面的 if 语句可以用以下赋值语句实现:

```
max=(a>b)?a:b;
```

其中,“(a>b)?a:b”是一个条件表达式,其含义是:若条件(a>b)成立,则条件表达式取值 a;否则,取值 b。

(1) 条件表达式的一般格式。

表达式 1? 表达式 2: 表达式 3



注意

条件运算符要求有三个操作对象,称为“三目运算符”(它是 C 语言中唯一的一个三目运算符)。条件表达式中的“表达式 1”、“表达式 2”、“表达式 3”缺一不可,其类型可以各不相同。

例如:

```
main(void)
{
    int x,y;
    x=80;
    y=x>70?100:0;
    printf("y=%d",y);
}
```

屏幕显示程序的运行结果为:

y=100

(2) 条件表达式的运算规则。

如果“表达式 1”的值为非 0 (即逻辑真),则运算结果等于“表达式 2”的值;否则,运算结果等于“表达式 3”的值。

条件表达式的流程图如图 4-4 所示。

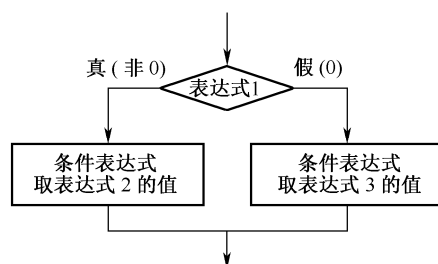


图 4-4 条件表达式的流程图

(3) 使用说明。

① 条件运算符的执行顺序。先求解表达式 1，若其值为真（非 0），则求解表达式 2，且整个条件表达式的值等于表达式 2 的值；若表达式 1 的值为假（0），则求解表达式 3，且整个条件表达式的值等于表达式 3 的值。如：

`max=(x>=0)?x:-x;` /*先进行条件运算后，再把条件表达式的值赋给变量 max */

② 运算符的优先级。条件运算符的优先级高于赋值运算符，但低于算术运算符和关系运算符。例如：

`max=(a>b)?a:b;` 等价于：`max=a>b?a:b;`

`a>b?a:b+1` 等价于：`(a>b)?a:(b+1)`

③ 条件运算符的结合方向。

条件运算符的结合方向为“从右到左”（即右结合性）。例如：

`a>b? a:c>d?c:d` 等价于：`(a>b)?a:(c>d?c:d)`

例 4-4 从键盘上输入一个字符，判别它是否为大写字母，如果是，则把它转换成小写字母输出；否则不转换，直接输出。

提示：字符在计算机内是用 ASCII 表示，大写字母的 ASCII 值正好比相应的小写字母小 32，因此，若能判断出所输入的字符为大写字母，则将该字符的 ASCII 值加上 32 后，即转换为相应的小写字母。

判断所输入的字符是否为大写字母的条件是：`ch>='A' && ch<='Z'` 为真。

程序如下：

`/*exam4-4*/`

`main(void)`

`{`

`char ch;`

`printf("Input a character: ");`

`scanf("%c",&ch);`

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

`printf("ch=%c\n",ch);`

`}`

若从键盘上输入一个字符 d，则屏幕显示程序的运行结果为：

Input a character: d ✓

ch=d

若从键盘上输入一个字符 D，则屏幕显示程序的运行结果为：

Input a character: D ✓

ch=d

4.2 switch 语句

if 语句一般用于处理两个分支的选择结构，当需要处理多个分支的选择结构时，需使用 if 语句的嵌套结构，if 语句的嵌套一般不宜超过 3 层，否则会影响程序的易读性。而 C 语言提供了能直接处理多个分支选择结构的 switch 语句，极大地方便了选择结构程序设计。

1. switch 语句的一般格式：

switch(表达式)

```
{case 常量表达式 1:语句 1;break;
case 常量表达式 2:语句 2;break;
.....
case 常量表达式 n:语句 n;break;
[default:语句 n+1;[ break;] ]
}
```

例如：

```
switch(grade)
{
case 'A':printf("85~100\n");break;
case 'B':printf("70~84\n");break;
case 'C':printf("60~69\n");break;
case 'D':printf("<60\n");break;
default: printf("error\n");
}
```

该 switch 语句的流程图如图 4-5 所示。

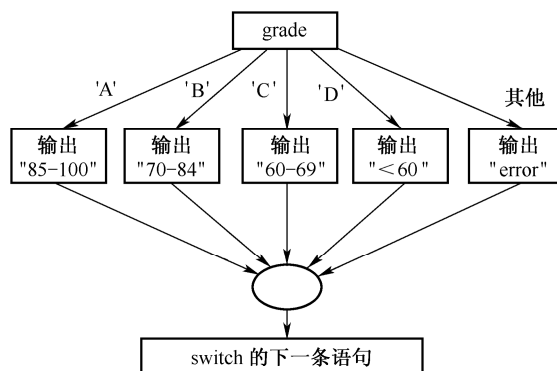


图 4-5 switch 语句的流程图



注意

case 后面为语句组（即包含多个语句）时，不需要加大括号。

2. switch 语句的功能

其功能包括以下两方面。

(1) 当 switch 后面“表达式”的值与某个 case 后面的“常量表达式”的值相同时, 就执行该 case 后面的语句(组); 当执行到 break 语句时, 则跳出 switch 语句, 转向执行 switch 语句的下一条。

(2) 如果没有任何一个 case 后面的“常量表达式”的值与 switch 后面“表达式”的值相同, 则执行 default 后面的语句(组)。然后, 跳出 switch 语句, 执行 switch 语句的下一条。

3. 使用说明

(1) switch 后面的“表达式”和 case 后面的“常量表达式”可以是整型、字符型和枚举型中的任意一种, 程序执行时将自动计算其值。

(2) 为了避免程序执行时出现自相矛盾的现象, 对 switch 后面的“表达式”的值只能有一种执行方案, 因此, 每个 case 后面“常量表达式”的值, 必须各不相同。

(3) case 后的语句中 break 语句不能缺。

case 后面的“常量表达式”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, 从此标号开始执行, 不再进行标号判断, 所以在每个 case 的语句(组)后面必须加上 break 语句, 以便结束 switch 语句。

**思考**

如果 case 中的所有 break 语句都省略不写, 如下面程序所示, 而 switch 后面的“表达式”的值为 B, 则该段程序输出的结果会怎样?

```
switch(grade)
{ case 'A':printf("85~100\n");
  case 'B':printf("70~84\n");
  case 'C':printf("60~69\n");
  case 'D':printf("<60\n");
  default: printf("error\n");
}
```

(4) 当每一个 case 语句后均有 break 语句时, 各 case 出现的先后次序不影响执行结果。default 语句总是放在所有的 case 语句后面, 它后面不需要写 break 语句。

(5) 多个 case 子句, 可共用同一语句(组)。如:

```
switch(grade)
{ case 'A':
  case 'B':
  case 'C': printf(">60\n"); break;
}
```

在变量 ch 的值为 A、B、C 三种情况下, 均执行相同的语句组“printf(">60\n"); break;”。

(6) 用 switch 语句实现的多分支结构程序, 完全可以用 if 语句或 if 语句的嵌套来实现。

例 4-5 从键盘上输入一个百分制成绩 score, 按下列原则输出其等级: $\text{score} \geq 90$, 等级为 A; $80 \leq \text{score} < 90$, 等级为 B; $70 \leq \text{score} < 80$, 等级为 C; $60 \leq \text{score} < 70$, 等级为 D; $\text{score} < 60$, 等级为 E。

```
/*exam4-5*/
main(void)
{
    int score, grade;
    printf("Input a score(0~100): ");
    scanf("%d", &score);
```

```

grade=score/10;          /*将成绩整除 10，得一整型值作为 switch 的表达式值*/
switch(grade)
{ case 10:
  case 9: printf("grade=A\n"); break;
  case 8: printf("grade=B\n"); break;
  case 7: printf("grade=C\n"); break;
  case 6: printf("grade=D\n"); break;
  case 5:
  case 4:
  case 3:
  case 2:
  case 1:
  case 0: printf("grade=E\n"); break;
  default: printf("The score is out of range!\n");
}
}

```

从键盘上输入百分制成绩为 95，则屏幕显示程序运行情况如下：

```

Input a score(0~100): 95↵
grade=A

```

4.3 选择结构程序设计举例

例 4-6 求任一元二次方程 $ax^2+bx+c=0$ 的解。 a 、 b 、 c 为方程的系数，其中 $a \neq 0$ 。

提示：求解一元二次方程的根有以下几种可能性：

- (1) 当 $a = 0$ 时，不能构成一元二次方程。
- (2) 当 $b^2-4ac=0$ 时，一元二次方程有两个相等的实根。
- (3) 当 $b^2-4ac>0$ 时，一元二次方程有两个不等的实根。
- (4) 当 $b^2-4ac<0$ 时，一元二次方程有两个共轭复根。

为了编程方便，设变量 $disc$ 表示一元二次方程判别式 b^2-4ac 。由于实数在计算机中运算和存储时，不可避免地会有一些微小误差，因此在判别 $disc$ （为实数）是否为零时，不能直接与零比较，正确的方法是判断其绝对值小于一个很小的常数，如采用方法： $if(fabs(disc) <= 1E-6)$ ，而不是： $if(disc == 0.0)$ 。

N-S 流程图如图 4-6 所示。

程序如下：

```

/*exam4-6*/
#include "math.h"
main(void)
{float a,b,c,disc,x1,x2,p,q;
  printf("Please input a,b,c= ");
  scanf("%f%f%f", &a, &b, &c);          /*输入一元二次方程系数*/
  printf("The equation: ");
  printf("%fx*x+%fx+%f=0",a,b,c);
  if(fabs(a) <= 1E-6)                  /*判断系数 a = 0*/

```

非二次方程	真	a=0		假
	真	b ² -4ac=0		假
	相等实根 -b/2a	真	b ² -4ac>0	
		假		
		$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$ $x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$	$p = -\frac{b}{2a}$ $q = \frac{\sqrt{-b^2 + 4ac}}{2a}$	
		两实根: x1、x2	复根: p+qi p-qi	

图 4-6 N-S 流程图

```

{ printf("is not quadratic\n"); exit(0); }          /*输出不能构成方程的信息后，退出*/
else
    disc=b*b-4*a*c;                               /*定义判别式Δ*/
    if (fabs(disc)<=1e-6)                           /*fabs()为求绝对值库函数，判断Δ==0*/
        {printf("has two equal root:\n");
        printf("x1=x2=%8.2f\n", -b/(2*a));          /*输出两个相等的实根*/
        }
    else
        if (disc>1e-6)                             /*判断Δ>0*/
            {x1=(-b+sqrt(disc))/(2*a);              /*求出两个不相等的实根*/
            x2=(-b-sqrt(disc))/(2*a);
            printf("has distinct real roots: \n");
            printf(" x1=%8.2f,x2=%8.2f\n", x1, x2);
            }
        else                                        /*Δ<0*/
            {p=-b/(2*a);                            /*求出两个共轭复根，其中p表示实部*/
            q=sqrt(fabs(disc))/(2*a);                /*q表示虚部*/
            printf("has complex roots:\n");
            printf("x1=%7.2f + %7.2f i\n", p, q);    /*输出两个共轭复根之一：x1*/
            printf("x2=%7.2f - %7.2f i\n", p, q);    /*输出两个共轭复根之二：x2*/
            }
    }
}

```

若输入方程系数 a、b、c 的值分别为 1、6、2，则屏幕显示程序的运行情况如下：

Please input a,b,c= 1,6,2✓

The equation: 1.000000x*x+6.000000x+2.000000=0 has distinct real roots:

x1= -0.35,x2= -5.56

若输入方程系数 a、b、c 分别为 1、2、3，则屏幕显示程序的运行情况如下：

Please input a,b,c= 1,2,3✓

The equation: 1.000000x*x+2.000000x+3.000000=0 has complex roots:

x1= -1.00+ 1.41i

x2= -1.00- 1.41i



思考

如果将系数 a、b、c 定义成整数，能否直接判断一元二次方程判别式 disc 是否等于 0？

例 4-7 已知某公司员工的保底薪水 salary 为 500，某月所接工程的利润 profit（整数）与利润提成的关系如下（计量单位：元）：

$\text{profit} \leq 1000$	没有提成；
$1000 < \text{profit} \leq 2000$	提成 10%；
$2000 < \text{profit} \leq 5000$	提成 15%；
$5000 < \text{profit} \leq 10000$	提成 20%；
$10000 < \text{profit}$	提成 25%。

编程实现：统计员工实际薪水。

提示：定义变量 salary 表示员工实际薪水，变量 profit 表示员工所接工程的利润，考虑到工程利润与利润提成的关系有五种情况，故采用多路分支结构编写程序。

为使用 switch 语句，必须将利润 profit 与提成的关系，转换成某些整数与提成的关系。分析本题可知，提成的变化点都是 1000 的整数倍（1000、2000、5000、……），如果将利润 profit 整除 1000，商用变量 grade 表示，则当：

$\text{profit} \leq 1000$	grade 可能为 0、1
$1000 < \text{profit} \leq 2000$	grade 可能为 1、2
$2000 < \text{profit} \leq 5000$	grade 可能为 2、3、4、5
$5000 < \text{profit} \leq 10000$	grade 可能为 5、6、7、8、9、10
$10000 < \text{profit}$	grade 可能为 10、11、12、……

为解决相邻两个区间的重叠问题，最简单的方法就是：利润 profit 先减 1（最小增量），然后再整除 1000 即可。

$\text{profit} \leq 1000$	grade 可能为 0
$1000 < \text{profit} \leq 2000$	grade 可能为 1
$2000 < \text{profit} \leq 5000$	grade 可能为 2、3、4
$5000 < \text{profit} \leq 10000$	grade 可能为 5、6、7、8、9
$10000 < \text{profit}$	grade 可能为 10、11、12、……

程序如下：

```
/*exam4-7*/
main()
{ long profit;
  int grade;
  float salary=500;
  printf("Input profit: ");
  scanf("%ld", &profit);
  grade=(profit-1)/1000;      /*将利润-1、再整除 1000，转化成 switch 语句中的 case 标号*/
  switch(grade)
  { case 0: break;           /*profit≤1000 */
    case 1: salary+=profit*0.1; break; /*1000<profit≤2000*/
    case 2:
    case 3:
    case 4: salary+=profit*0.15; break; /*2000<profit≤5000*/
    case 5:
    case 6:
```

```
case 7:
case 8:
case 9: salary+=profit*0.2; break;    /*5000<profit≤10000*/
default: salary+=profit*0.25;        /*profit>10000*/
}
printf("salary=%.2f\n", salary);
}
```

若从键盘上输入员工所接工程的利润为 2600, 则屏幕显示程序的运行情况如下:

```
Input  profit: 2600 ✓
salary=890.00
```

习题四

一、填空题

1. 写出下列命题的 C 语言描述。

- (1) x 小于 y 或小于 z _____,
- (2) x 和 y 都大于 z _____,
- (3) x 或 y 中有一个小于 z _____,
- (4) x 是偶数_____。

2. 根据以下 if 语句写出与其功能相同的 switch 语句 (x 的值在 0~100 之间。)

if 语句:

```
if(a<40) b=1;
else if(a<50) b=11;
else if(a<60) b=111;
else if(a<70) b=1111;
else if(a<80) b=11111;
```

switch 语句:

```
switch(_____)
{
    _____b=1;break;
    case 4:b=11;break;
    case 5:b=111;break;
    case 6:b=1111;break;
    _____b=11111;
}
```

3. 设有程序段如下:

```
switch(rank)
{
    case 1:printf("PERFECT! \n");
    case 2:printf("GOOD! \n");
    case 3:printf("JUST SOSO! \n");
    case 4:printf("BAD! \n");
    default:printf("IMPOSSIBLE! \n");
}
```

若 rank 的值为 3，则输出结果是_____。

4. 把以下两条 if 语句合并成一条 if 语句_____。

```
if(m<=n) x=100;
else y=50;
if(m>n) printf("the result is y=%d\n",y);
else printf("the result is x=%d\n",x);
```

5. 阅读下面程序，将问题的答案填入对应的划线处。

```
#include <stdio.h>
main()
{   int  a, b, m, n;
    scanf("%d %d",&a,&b);
    m=1;
    n=1;
    if(a>0) m=m+n;
    if(a<b)n=2*m;
    else if(a==b) n=5;
        else n=m+n;
    printf("m=%d  n=%d\n", m, n);
}
```

问题:

①当输入: -1 -2<回车>, 程序的运行结果为_____。

②当输入: 1 0<回车>, 程序的运行结果为_____。

③为了输出 n=4, 变量 a 和 b 应具备条件_____。

6. 补足程序语句, 使程序实现计算一元二次方程 $ax^2+bx+c=0$ 的根的功能。

```
#include <math.h>
main()
{   float  a,b,c,_abs,delta,m,n1,n2;
    printf("enter  a, b, c:  ");
    scanf("%f %f %f",&a,&b,&c);
    if(_____)
        if(_____) printf("warning!input  error\n");
    else printf("the equation has a single root is %f\n",-c/b);
    else
    {   delta=b*b-4*a*c;
        m=2*a;
        n1=-b/(2*a);
        _abs=abs(delta);
        n2=sqrt(_abs)/m;
        if(_____)
            printf("the equation has two complex roots\n real
            part=%f  image  part=%f\n",n1,n2);
        else
            printf("the equation has two real roots\n root1=%f
            root2=%f\n", n1+n2, n1-n2);
    }
}
```

7. 以下程序的运行结果是_____。

```
main()
{   int m=1,n=0;
    switch(m)
    {   case 1:
        switch(n)
        {   case 0:printf("It is a dog\n");break;
            case 1:printf("It is a cat");break;
        }
        case 2:printf("No meaning\n");
    }
}
```

二、选择题

1. 两次运行下面的程序，如果从键盘上分别输入 12 和 4，输出结果是（ ）。

```
main()
{
    int m;
    scanf("%d",&m);
    if(m++>11) printf("%d",m);
    else printf("%d\n", m--);
}
```

- A. 13 和 5 B. 12 和 5 C. 12 和 6 D. 13 和 6
2. 如下程序

```
main()
{   int a=5, b=2, c=0;
    if(c<0)
        if(b<0) a=3;
        else a=4;
    printf("%d\t",a);
    c=a;
    if((c=b)<0) a=3;
    else if(b==0) a=5;
        else a=7;
    c=a-c;
    printf("%d\t",a);
    printf("%d\t", c);
}
```

运行结果是（ ）。

- A. 5 7 2 B. 3 7 2 C. 5 7 5 D. 3 5 7
3. 以下哪一个是判断字符型变量 ch 为小写字母的表达式（ ）。
- A. (ch>=a)AND(ch<=z) B. (ch>=a)&(ch<=z)
- C. (ch>=a)&&(ch<=z) D. (ch>=a)|| (ch<=z)
4. 假定所有变量均已正确说明，下列程序段运行后 m 的值是（ ）。

```

a=b=0;c=3;m=24;
if(!a) m=-1;
else if(b);
if(c) m=3;
else m=4;

```

A. -1 B. 3 c. 24 D. 3

5. 如下程序

```

main()
{
    int m, n=1;
    if(n!=0) m=6;
    printf("%d\t", m);
    if(n==0) m=12;
    else m=7;
    printf("%d\t\n", m);
}

```

执行结果是 ()。

A. 6 12 B. 6 0 C. 6 5 D. 6 7

6. 若有以下函数关系

$x < 0$, 则 $y = 3x * x$

$x > 0$, 则 $y = 5x$

$x = 0$, 则 $y = 3x * x + 4$

下面程序段能正确表示以上关系的是 ()。

A.

```

y=3*x*x;
if(x!=0)
else if(x>0) y=5*x;
else y=3*x*x+4;

```

C.

```

if(x>=0)
if(x>0) y=5*x;
else y=3*x*x+4;
else y=3*x*x;

```

B.

```

y=3*x*x;
if(x<=0)
else if(x==0) y=3*x*x+4;
else y=5*x;

```

D.

```

y=3*x*x+4;
if(x<=0)
else if(x<0) y=3*x*x;
else y=5*x;

```

三、编程题

1. 编程打印所有各个位数的立方和等于该整数的三位数。
2. 请编程, 利用公式 $e = 1 + 1/1! + 1/2! + 1/3! + \dots + 1/n!$ 求出 e 的近似值, n 的值由用户输入。
3. 输入三角形边长, 请编程序计算三角形面积。
4. 用户输入圆的半径以及一个选项整数, 当整数为 0 时, 求圆面积; 整数为 1 时, 求圆周长; 整数为 2 时, 则既求面积也求周长。

实训四

一、实训目的

1. 掌握关系运算符、关系表达式和逻辑运行符、逻辑表达式在程序设计中的应用。
2. 熟练掌握 if 语句和 switch 语句的应用。
3. 掌握选择结构的程序设计方法。

二、实训内容

1. 编程实现以下分段函数，并上机运行。

$$y = \begin{cases} x & \text{当 } x < 0 \text{ 时} \\ 2x+1 & \text{当 } 0 \leq x < 10 \text{ 时} \\ 4x-3 & \text{当 } x \geq 10 \text{ 时} \end{cases}$$

要求：

- (1) 用 scanf 函数输入 x 的值；
- (2) 输入不同的 x 值（分别为 $x < 0$ 、 $0 \leq x < 10$ 、 $x \geq 10$ ），测试输出 y 值是否正确。

2. 输入一个学生的百分制成绩，要求输出相应的等级评价。90 分以上的为'A'，80 至 89 分为'B'，70 至 79 分为'C'，60 至 69 分为'D'，60 分以下为'E'。

要求：

- (1) 分别用 if 语句和 switch 语句编程实现。
- (2) 分别输入分数为正常范围 0~100 以内的值、大于 100 的值、小于 0 的值，测试程序的正确性。当输入分数不在 0~100 范围内时不应给出等级，而是给出提示“输入数据错”。

3. 编程实现：输入一个不多于 5 位的正整数：

- (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：原数据为 54321，则应输出结果为 12345。

要求使用以下测试数据：

- 要处理的数为 1 位正整数；
- 要处理的数为 2 位正整数；
- 要处理的数为 3 位正整数；
- 要处理的数为 4 位正整数；
- 要处理的数为 5 位正整数。

除此之外，程序还应当对不合法的输入作必要的处理。如以下情形：

- 输入负数；
- 输入的数超过 5 位（如 123456）。