第3章 顺序结构程序设计

3.1 学习要求

顺序结构是最基本的程序结构，顺序结构程序设计是最简单的程序设计，程序的执行顺序是从上到下逐条执行语句。

本章学习要求：

(1) 主要掌握C语言的运算符与表达式，表达式语句、赋值语句、空语句和复合语句的用法。

(2) 数据输入输出的概念及在C语言中的实现方法，掌握字符数据的输入输出方法，重点掌握格式输入输出函数的使用。

(3) 学会顺序结构程序设计的方法和步骤，能够进行顺序结构的程序设计。

3.2 基本知识

3.2.1 运算符与表达式

1. 算术运算符及其用法

(1) 算术运算符。包括5种：+(加/取正)、-(减/取负)、*(乘)、/(除)、%(求余数)。

(2) 使用方法。关于除法运算：C语言规定，两个整数相除，其商为整数，小数部分被舍弃。例如，5/2= 2，-5/2.0 = -2.5。

关于求余数运算：要求两侧的操作数均为整型数据，否则出错。例如，7%4=3，而7.2%3是不合语法规则的，如程序中出现这样的表达式，就会出错。

2. 自增、自减运算符及其用法

(1) 自增、自减运算符：

- ++：作用是使变量的值加1。
- --：作用是使变量的值减1。

(2) 使用方法：

- ++i,--i 用法：先加(减)1,再使用。
- i++,i--用法：先使用,再加(减)1。

例如：

```
i=2;
j=++i;      //i 的值变成 3,然后赋给 j,所以,i,j 的值都是 3
printf("i=%d,j=%d\n",i,j);
```

输出结果为：

```
i=3, j=3
```

再如：

```
i=2;
j=i++;      //i 的值先赋给 j,j=2。然后,i 的值变成 3,所以,i 的值是 3
printf("i=%d,j=%d\n",i,j);
```

输出结果为：

```
i=3, j=2
```

注意：自加运算符“++”与自减运算符“--”是单目运算符,运算对象必须是变量。

3. 赋值运算符及其用法

(1) 赋值运算符。赋值符号“=”就是赋值运算符,赋值运算符的一般形式为：

```
变量=表达式
```

注意：赋值运算符左边必须为变量,赋值运算是把赋值运算符右边表达式的值赋给左边变量。

例如,x=y+2 是合法的,而 x+2=y 是不合法的。

(2) 复合赋值运算符。在赋值运算符(=)之前加上算术运算符(+、-、*、/、%)构成复合赋值运算符。

掌握复合赋值表达式转化为赋值表达式的方法。

例如：

```
n+=100  等价于  n=n+100
```

再如：

```
x%=y+2  等价于 x=x%(y+2)
```

4. 算术运算符和赋值运算符的优先级和结合性

算术表达式中,可使用多层圆括号,但括号必须配对。

算术运算时,先乘、除、求余数,后加、减。括号优先。同一级别从左到右。

算术运算符优先级较高于赋值运算符。多数运算符左结合性，而赋值运算符结合性自右向左。

例如：

```
a=(b=5)
```

先执行 b=5，b 的值等于 5，然后再执行 a=b，a 的值也等于 5。

注意：根据赋值运算符自右向左的结合性，a=(b=5)等价于 a=b=5。

3.2.2 类型转换

1. 自动转换

在不同类型数据的混合运算中，由系统自动实现转换，由少字节类型向多字节类型转换。

不同类型数据混合运算时的转换规则如图 3.1 所示，图中箭头指示方向为数据转换方向。

例如，10+'a'+1.5-45.67*'b'运算，运算结果为 double 型。

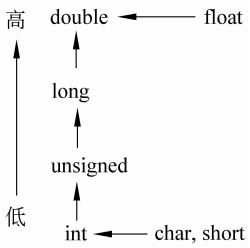

图 3.1 不同类型数据的转换规则

2. 强制转换

由强制转换运算符完成转换。一般形式为：

```
(类型名)(表达式)
```

例如：

```
float x=3.6;
int i;
i=(int)x;
printf("x=%f  i=%d\n",x,i);
```

则输出结果为：

```
x=3.600000  i=3
```

x 的类型仍为 float 型，i 的类型为 int 型。

3.2.3 输入与输出

C 语言输入输出都是通过调用函数实现的，这些函数包含在头文件 stdio.h 中，所以在调用输入输出函数时，要在文件头加上预处理命令：

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

1. printf 函数(格式输出函数)

printf 函数一般形式为:

```
printf(格式控制,输出列表)
```

输出控制是用一对双引号引起来的,包含格式说明和普通字符。输出列表包含若干输出项。例如:

```
int a=25,b=136;
printf("a=%d  b=%d\n",a,b);
```

则输出结果为:

```
a=25 b=136
```

注意:

(1) 格式说明中,%d 对应整型,%f 对应单精度实型,%c 对应字符型,%s 对应字符串型。

(2) 可在%和格式字符之间加格式控制字符来控制数据输出格式。实数输出可控制数据宽度和小数位数。

2. scanf 函数(格式输入函数)

scanf 函数一般形式为:

```
scanf(格式控制,地址表列)
```

地址表代表输入项,要求带取地址符 &,表示需要读入的变量的地址,而不是变量本身。

注意:使用 scanf 函数时,键盘输入内容必须和格式控制一致,否则会出现错误。"scanf("%d, %d, %d", &a, &b, &c);"中"%d, %d, %d"之间有逗号,在输入数据时也要加逗号,如果去掉逗号,输入时就不用逗号,而用空格或 将各个数据隔开。例如:

```
scanf("%d,%d,%d", &a, &b, &c);
```

可以输入为:

```
3, 4, 5
```

不能输入:

```
3 4 5
```

3. getchar 函数(单个字符输入函数)

getchar 函数的功能是从键盘输入一个字符。

getchar 函数一般形式为:

```
getchar()
```

注意：getchar 函数只能接收单个字符，输入数字也按字符处理。输入多于一个字符时，只接收第一个字符。

4. putchar 函数（单个字符输出函数）

putchar 函数的功能是在显示器上输出单个字符。

putchar 函数一般形式为：

```
putchar(字符常量或字符变量)
```

例如：

```
putchar('A');     // 输出大写字母 A
putchar('\n');    // 换行。对控制字符则执行控制功能，不能在屏幕上显示
```

3.2.4 宏常量与宏替换

define 用于定义预编译时处理的宏，编译时只进行简单的字符替换。

例如：

```
#define PI 3.14
```

PI 称为符号常量。

3.2.5 典型例题

【例 3-1】 从键盘输入两个整数，求它们的和、差、积、商以及余数，然后输出结果。

问题解析：

（1）先定义需要的变量并确定合适的数据类型。除了定义两个整数类型变量 a、b 外，还要定义两个数的和、差、积、商和余数的变量。

想一想，这 5 个变量分别要定义为什么类型？

（2）从键盘输入任意两个整数 a、b。

（3）计算这两个数的和、差、积、商以及余数。

（4）输出计算结果。

编写程序如下：

```
#include <stdio.h>
int main()
{
    int a,b,he,cha,cj,ys;
    float shang;
    printf("input a,b:\n");
```

```
    scanf("%d,%d",&a,&b);
    he=a+b;
    cha=a-b;
    cj=a*b;
    shang=(float)a/b;
    ys=a%b;
    printf("he=%d cha=%d cj=%d shang=%.2f ys=%d\n",he,cha,cj,shang,ys);
    return 0;
}
```

运行结果：

```
input a,b:
7,3
he=10 cha=4 cj=21 shang=2.33 ys=1
```

注意事项：

(1) 求两个数的商时，要定义 shang 变量类型为实数类型。

(2) 因为 a 和 b 都是 int 类型，a/b 的结果也是整数，可以把 a 变量的值转换为 float 类型，(float)a/b 的运算结果为实数类型。

(3) 输出商的结果时，要用%f，其他整型变量输出时，对应格式说明符为%d。

【例 3-2】 从键盘输入一个 5 位数，求该数个位、十位、百位、千位、万位上的各数的和，并输出计算结果。

问题解析：

解决此问题的关键是求出该数的个位、十位、百位、千位、万位上的数，然后累加求和。

编写程序如下：

```
#include <stdio.h>
int main()
{
    int n, sum;
    int ge,shi,bai, qian,wan;    //个、十、百、千、万位数字变量
    printf("请输入一个整数(10000-99999):\n");
    scanf("%d",&n);
    wan=n/10000 ;                //求万位数
     qian =n/1000%10;            //求千位数
     bai=n/100%10 ;              //求百位数
     shi=n/10%10  ;              //求十位数
    ge=n%10 ;                    //求个位数
    sum=ge+shi+bai+qian+wan;
    printf("sum=%d\n",sum);
    return 0;
}
```

运行结果：

```
请输入一个整数(10000-99999)
12345
sum=15
```

思考：求个位、十位、百位、千位、万位的方法不唯一，读者如何灵活应用运算符 / 和%，用别的方法计算？

3.3 等级考试知识

3.3.1 等级考试考点

1. C 算术运算符和赋值运算符

本考点不会以单独形式考察，而是通过运算符来实现程序特定功能，属于简单识记内容，考核概率约为 30%。

2. C 自增自减运算符

本考点属于难点，在后面的 for 语句中经常用到，考核概率约为 20%。

3. 不同类型数据间的转换与运算

本考点在选择题、操作题中都有出现。要求能够理解不同类型数据间的转换与及在赋值运算时的运算规则。本考点属于难点，考核概率约为 20%。

4. 表达式和语句

表达式和语句是 C 语言程序的构成单位，贯穿在整个 C 语言的学习中，需要在学习中体会 C 语句的用法。本考点最基础、最重要，常与后面章节内容结合考核。

5. 用 printf 函数实现格式输出控制

在 C 程序中，结果输出必不可少，而 printf 函数可以根据要求输出各种形式，考核概率 100%，难点是按指定格式输出不同类型的若干数据。

6. 用 scanf 函数实现格式输入

考核 scanf 函数的用法。根据输入格式判断输出内容，或根据输出内容书写正确的输入格式控制，考核概率 100%。

3.3.2 等级考试例题

选择题

1. 已知 x、y、n 为 int 类型，y＝2，以下为合法 C 语言赋值语句的是(　　)。

A. x＝y＋5　　B. x＝n％2.5　　C. x＋n＝y　　D. x＝5＝4＋3

【解析】 本题考察赋值运算符(＝)和求余数运算符(％)的用法。赋值运算符左边必须为变量，赋值运算是把赋值运算符右边常量、变量或表达式的值赋给左边的变量，选项 D 错误，A 正确。“％”运算符要求两侧的操作数均为整型数据，否则出错，选项 B 错误。

【答案】 A

2. 以下程序的输出结果是(　　)。

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

A. x＝3，y＝3　　B. x＝4，y＝3　　C. x＝4，y＝4　　D. x＝3，y＝4

【解析】 本题考察自增自减运算符用法。x＋＋的运算顺序是先使用，后增值。x 先输出 3，然后，x 的值加 1 变成 4。＋＋y 的运算顺序是先增值，后使用。y 的值先加 1 变成 4，然后输出 y 的值 4。

【答案】 D

3. 若有定义：“int x＝3；float a＝3.5，b＝6.2；”，则表达式(int)a％2＋x/(int)b 的值为(　　)。

A. 1.0　　B. 1　　C. 2　　D. 2.0

【解析】 本题考察强制类型转换和算术运算符的优先级。(int)a 的值为 3，(int)b 的值为 6，(int)a％2 的值为 1，x 与(int)b 都是 int 类型，运算结果仍为 int 类型，x/(int)b 的值为 0，(int)a％2＋x/(int)b 的值为 1，选项 A、C、D 错误，B 正确。

【答案】 B

4. 有以下程序段：

```
int x=123;
double y=3.141593;
printf("%2d,%8.4f\n",x,y);
```

输出结果(□代表空格)是(　　)。

A. 12，□□3.1416　　B. 123，□□□3.1416

C. 123，□□3.1416　　D. 12，3.1415□□

【解析】 本题考察 printf 函数格式输出。“％2d”输出整数。指定宽度 2 小于实际宽度 3，按实际宽度输出，输出“123”。

“%8.4f”输出实数。“8”为输出实数位数(包括小数点),“4”为输出小数点后位数(最后一位4舍5入),指定宽度8大于实际宽度6,左补两个空格,输出“□□3.1416”,选项A、B、D错误,C正确。

【答案】 C

5. 有以下程序段:

```
int x=12;
double y=3.141593;
printf("%-4d,%-8.4f\n",x,y);
```

输出结果是(　　)。

A. 12□□,□□3.1416　　B. 12,□□3.1416

C. □□12,□□3.1416　　D. 12□□,3.1416□□

【解析】 本题考察printf函数的格式输出。“%-4d”输出整数。指定宽度4大于实际宽度2,且宽度4前面是“-”号,右补两个空格,输出“12□□”。

“%-8.4f”输出实数。“8”为输出实数位数(包括小数点),“4”为输出小数点后位数(最后一位4舍5入),指定宽度8大于实际宽度6,且宽度8前面是“-”号,右补两个空格,输出“3.1416□□”,选项A、B、C错误,D正确。

【答案】 D

6. 有以下程序:

```
#include <stdio.h>
main()
{
    int x,y;
    scanf("%2d%3d",&x,&y);
    printf("%d,%d\n",x,y);
}
```

输入1234567,输出结果是(　　)。

A. 12,34567　　B. 123,123　　C. 12,345　　D. 12,3456

【解析】 本题考察scanf函数的格式输入。“%2d”取两位整数12,x的值为12。“%3d”取三位整数345,y的值为345。选项A、B、D错误,C正确。

【答案】 C

3.4　顺序结构编程实验

3.4.1　实验目的

(1) 熟练掌握单个字符输入输出函数的使用。

(2) 熟练掌握格式输入与格式输出函数的使用。

(3) 使用顺序结构解决几个简单的实际问题。

(4) 学习简单的程序调试方法,能够对常见错误进行处理。

3.4.2 实验内容

1. 从键盘输入梯形的上底 a、下底 b 和高 h,然后根据梯形面积公式求面积 s,最后输出结果。

要求:输出要有文字说明,计算结果取两位小数。

提示:可先输入梯形的上底 a、下底 b 和高 h,然后根据梯形面积公式计算求面积 s,最后输出结果。

请填空,把程序补充完整,并分析运行结果。

```
#include <stdio.h>
int main()
{    【  1  】 a,b,h,s;                          //定义变量
     printf("请输入梯形的上底、下底和高 a,b,h\n");
     scanf("%f,%f,%f",&a,&b,&h);               //从键盘输入 a,b,h
     s=【  2  】;                               //计算梯形的面积
     printf("梯形的面积 s=%.2f\n",【  3  】); //输出计算结果
     return 0;
}
```

2. 从键盘输入圆半径 r,求圆周长 L,圆面积 S1、球表面积 S2、球体积 V。

提示:圆周率 π 不能作为变量名,可以宏定义:#define PI 3.14。

【输入形式】

用 scanf 语句输入 r

【输出形式】

输出时要有文字说明,小数点后取 2 位小数。

【样例输入】(下画线部分表示输入内容)

```
Input r:1.5
```

【样例输出】

```
L=9.42 S1=7.07 S2=28.26 V=10.60
```

【样例说明】 样例输出数据之间有一个空格隔开。

3. 输入并运行以下程序:

```
#include <stdio.h>
int main()
{
    int i, j,m,n;
```

```
    i=5;
    j=10;
    m=i++;
    n=j++;
    printf("i=%d,j=%d,m=%d,n=%d\n",i,j,m,n);
    i=5;
    j=10;
    m=++i;
    n=++j;
    printf("i=%d,j=%d,m=%d,n=%d\n",i, j,m,n);
    return 0;
}
```

在上机前先分析程序,写出结果,然后与上机结果对照。

运行结果是:_____

_____

4. 输入并运行以下程序:

```
#include <stdio.h>
int main()
{
      char ch='A';
     float x=12.34;
     printf("ch=%c ch=%d\n",ch,ch);
     printf("x=%f,x=%.2f,x=%-7.2f,x=%7.2f,x=%3.2f\n",x,x,x,x,x);
     return 0;
}
```

在上机前先分析程序,写出结果,然后与上机结果对照。

运行结果是:_____

_____

5. 输入下面程序并运行分析结果。

```
#include <stdio.h>
int main()
{   char c1,c2,c3;
    c1=getchar();
    c2=getchar();
    c3=getchar();
    putchar(c1);
    putchar(c2);
    putchar(c3);
    putchar('\n');
    return 0;
}
```

分析本题程序的输出结果，解释 getchar 函数的工作过程。

注意：回车符也是字符。

【样例输入 1】

```
xyz
```

【样例输出 1】

```
xyz
```

【样例输入 2】

```
x
yz
```

【样例输出 2】

```
x
y
```

6. 从键盘依次输入某同学 3 门课的成绩，要求成绩变量用浮点数，计算并输出这 3 门课的平均分。

要求：平均分输出保留两位小数。

请填空，把程序补充完整，并分析运行结果。

```
#include <stdio.h>
int  main()
{  【  1  】 cj1,cj2,cj3,aver;
   printf("Input a,b,c: ");
   scanf("%f,%f,%f,%f",&cj1,&cj2,&cj3);
   aver=【  2  】;
   printf("aver=%.2f\n",【  3  】);
   return 0;
}
```

思考：如果计算 3 个整数的平均数，如何修改程序？

7. 鸡兔同笼问题：已知鸡兔总头数为 h，总脚数为 f，求鸡兔各多少只？

提示：令鸡 x 只，兔 y 只，有

```
x+y=h
2x+4y=f
```

找到求 x 和 y 的具体公式：

```
x=(4h-f)/2
y=(f-2h)/2
```

【样例输入】(下画线部分表示输入内容)

```
Input h:5
```

```
Input f:14
```

【样例输出】

```
x=3 y=2
```

【样例说明】 样例输出数据之间有一个空格隔开。

8. 下面程序功能是交换任意两个整数 a 和 b 的值。

提示：要用一个中间变量 t 来实现 a 和 b 的值交换。

请填空，把程序补充完整，并分析运行结果。

```
#include <stdio.h>
int main()
{
    int a,b,t;
    scanf("%d",&a);
    scanf("%d",&b);
    printf("a=%d,b=%d\n",a,b);
    【  1  】;【  2  】; 【  3  】;    //交换 a,b 的值
    printf("the result of swap is:");
    printf("a=%d,b=%d\n",a,b);
    return 0;
}
```

9. 下面定义的宏实现两个参数值的互换。在主函数中调用此宏分别完成两个整数、实数的互换，输出交换后的值。

```
#include <stdio.h>
#define change(a,b)  t=b;b=a;a=t;  //带参数的宏定义
void main()
{
    int a,b,t;
    printf("Enter integer a ,b:");
    scanf("%d,%d",&a,&b);
    change(a,b);
    printf("Now a=%d  b=%d\n",a,b);
}
```

输入并运行该程序，观察运行结果，体会宏定义实现的过程。

习　　题

一、选择题

1. 若有定义语句“int a，b，c;”，以下选项中赋值语句正确的是(　　)。

A. a=b+1;　　B. a=2=1;　　C. a+2=c ;　　D. a+b=c;

2. 设有定义语句“int x=2;”,以下表达式中,其值不为 6 的是(　　)。

A. x*=x+1　　B. 2*(x++)　　C. x*=(1+x)　　D. 2*x+2

3. 有以下程序段:

```
int x=12;
double y=3.141593;
printf("%d%8.6f",x,y);
```

输出结果是(　　)。

A. 123.141593　　B. 12 3.141593　　C. 12, 3.141593　　D. 123.1415930

4. 已知“int k,m=1;”,执行语句“k=m++;”后,k 的值是(　　)。

A. -1　　B. 0　　C. 1　　D. 2

5. 若有定义“int x=4;”,则执行语句“x += x *= x + 1;”后,x 的值为(　　)。

A. 5　　B. 20　　C. 40　　D. 无答案

6. 若有定义“float a=3.0, b=4.0, c=5.0;”,则表达式 1/2*(a+b+c)的值为(　　)。

A. 6.0　　B. 6　　C. 0.0　　D. 无答案

7. 以下程序段的输出结果是(　　)。

```
int a=1234;
printf("%2d\n",a);
```

A. 12　　B. 34　　C. 1234　　D. 提示出错,无结果

二、阅读程序题

1. 有以下程序:

```
#include <stdio.h>
int main( )
{
    float x=2.5;
    int y;
    y=(int)x;
    printf("x=%f,y=%d",x,y);
    return 0;
}
```

程序运行后,输出结果是:_____

2. 有以下程序:

```
#include <stdio.h>
int main( )
{
    char ch1,ch2,ch3,ch4;
```

```
    scanf("%c%c",&ch1,&ch2);
    ch3=getchar();
    ch4=getchar();
    printf("%c%c%c%c\n",ch1,ch2,ch3,ch4);
    return 0;
}
```

输入：

```
12 <回车>
34<回车>
```

程序运行后，输出结果是：_____

第4章 选择结构程序设计

4.1 学习要求

选择结构是程序设计的三种基本结构之一，选择结构程序不是按程序的书写顺序依次执行，而是根据条件有选择地执行其中的程序段，根据选择的分支不同，选择结构可分为：单分支选择结构、双分支选择结构和多分支选择结构。

本章学习要求：

(1) 掌握 C 语言中关系运算符和关系表达式、逻辑运算符和逻辑表达式、条件运算符和条件表达式的用法。

(2) 掌握 if 语句和 switch 语句的使用方法。

(3) 掌握选择结构程序设计的方法和步骤。

4.2 基本知识

4.2.1 运算符与表达式

1. 关系运算符与关系表达式

(1) 关系运算符：>(大于)、<(小于)、>=(大于等于)、<=(小于等于)、==(等于)、!= (不等于)。

(2) 优先级和结合性：(>、<、>=、<=)优先于(==、!=)，结合方向为从左向右。

关系运算符的优先级低于所有算术运算符，高于赋值运算符。

例如：

```
c>a+b      等价于      c> (a+b)
a>b!=c     等价于       (a>b) !=c
a=b>c      等价于      a= (b>c)
```

(3) 关系表达式。关系表达式是用关系运算符将两个操作数连接起来的表达式。关系表达式常用于条件判断，关系表达式的值为真表示条件成立，关系表达式的值为假表示

条件不成立，在C语言中用非0表示“真”，用0表示“假”。在关系运算过程中，计算机将所有非0数值均看作“真”，0看作“假”。

例如：

```
int a=3,b=2,c=1,d,f;
a>b       //表达式值1
(a>b)==c  //表达式值1
b+c<a     //表达式值0
'a'>0     //表达式值1
```

注意：

(1) 等号和赋值号的区别，例如a=10和a==10是不同的，前者是把赋值号右边的10赋值给赋值号左边的变量a，后者是读取等于号左边的变量a的值与右边的10进行等值比较。

(2) 应避免对实数作相等或不相等的判断。

2. 逻辑运算符与表达式

(1) 逻辑运算符：&&(逻辑与)、||(逻辑或)、!(逻辑非)。

(2) 优先级和结合性。

- 优先级为：!(逻辑非)→&&(逻辑与)→||(逻辑或)。
- 逻辑运算符和其他运算符之间的优先级关系可表示为：!(逻辑非)优先于算术运算符、关系运算符，算术运算符和关系运算符优先于&&(逻辑与)和||(逻辑或)，&&(逻辑与)和||(逻辑或)优先于赋值运算符。
- 结合方向为：&&(逻辑与)和||(逻辑或)从左向右，!(逻辑非)从右向左。

例如：

```
a<=x && x<=b   等价于    (a<=x) && (x<=b)
a>b&&x>y       等价于    (a>b)&&(x>y)
!a||a>b        等价于    (!a)||(a>b)
```

(3) 逻辑表达式。用逻辑运算符连接操作数组成的表达式称为逻辑表达式。参与逻辑运算的对象是逻辑量，在C语言中任意表达式都可以作为逻辑量来处理，具体处理规则是表达式的值不为0，则认为真，用1表示；表达式的值为0，则认为假，用0表示。逻辑运算的结果与关系运算的结果一样，也是逻辑量。

例如，设有

```
a=4;b=5;
```

则：

```
!a          表达式的值为0
a&&b        表达式的值为1
a||b        表达式的值为1
!a||b       表达式的值为1
```

```
4&&0||2        表达式的值为 1
```

3. 条件运算

(1) 条件运算符(？：)。C 语言提供了一种条件运算符和条件表达式,条件运算符(？:)是 C 语言唯一的三目运算符,它连接 3 个运算分量,条件表达式的一般形式为:

```
表达式 1 ?表达式 2 : 表达式 3
```

其执行过程为:先计算出表达式 1 的值,如果表达式 1 的值为“真”,则整个条件表达式的值为表达式 2 的值,否则整个条件表达式的值为表达式 3 的值。

(2) 优先级和结合性。

- 优先级为:条件运算符优于赋值运算符,比算术、关系、逻辑运算符低。
- 结合方向:自右向左。

4.2.2 选择结构

C 语言中实现选择结构的控制语句有两种:条件分支语句 if 和开关分支语句 switch。

1. 单分支结构

if 语句的单分支格式为:

```
if (表达式 P)
    语句 1
```

2. 双分支结构

if 语句的双分支结构的格式为:

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

3. if 语句的嵌套

if 语句的双分支结构中,语句 1 和语句 2 本身又可以是一个 if 语句,这就是 if 语句的嵌套。下面是常见的嵌套结构:

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

```
    ⋮
else if(表达式 m)
    语句 m
    else 语句 n
```

为了保证嵌套的正确性,在使用 if 语句进行嵌套时,一定要注意 if 与 else 的配对关系。在嵌套的 if-else 语句中,else 总是与上面的离它最近的尚未配对的 if 配对。

4. 开关语句 switch

switch 语句是 C 语言中提供的设计多分支选择结构的语句,又称为多分支结构的开关语句。

switch 语句的一般格式为:

```
switch(表达式)
{
    case 常量 1:[语句序列 1] [break;]
    case 常量 2:[语句序列 2] [break;]
    ⋮
    case 常量 n:[语句序列 n] [break;]
    [default:语句序列 n+1] [break;]
}
```

执行过程:计算表达式的值,与常量表达式的值比较,等于第 i 个值时,顺序执行语句序列 i、i+1、…、n,若执行完一个分支后不再执行下一个分支,则需要用 break 语句结束,跳转到其所在的 switch 语句后继续执行其他语句,若与所有常量表达式值都不相等,执行语句序列 n+1。

注意:

(1) switch 语句中表达式可以是任意类型,但运算结果必须为整型或字符型。

(2) case 常量 i:等价于语句标号,计算出的表达式值等于哪个语句标号,就从哪个位置开始顺序向下执行语句序列。因此,语句位置影响运行结果。

(3) switch 语句可以嵌套,break 语句只跳出它所在的 switch 分支。

(4) case 与常量表达式之间必须有空格。

4.2.3 典型例题

【例 4-1】 有 3 个整数 x、y、z,从键盘输入,输出其中的最大数。

方法 1:程序流程图如图 4.1 所示。

编写程序如下:

```
#include <stdio.h>
void main()
{
```

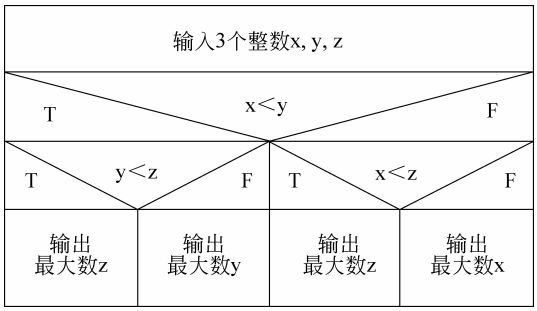

图 4.1　求三个数中的最大数程序流程图

```
    int x,y,z;
    printf("请输入整数 x,y,z:");
    scanf("%d,%d,%d",&x,&y,&z);
    if(x<y)
        if(y<z)
            printf("最大数为:%d\n",z);
        else
            printf("最大数为:%d\n",y);
    else if(x<z)
            printf("最大数为:%d\n",z);
        else
            printf("最大数为:%d\n",x);
}
```

运行结果：

```
请输入整数 x,y,z: 12,88,56
最大数为: 88
```

方法 2：使用条件表达式，可以使程序更简明、清晰。

编写程序如下：

```
#include <stdio.h>
main()
{
    int x,y,z,t,max;
    printf("请输入整数 x,y,z:");
    scanf("%d,%d,%d",&x,&y,&z);
    t=(x>y)? x:y;
    max=(t>z)? t:z;
    printf("3 个数中的最大数为:%d\n",max);
}
```

运行结果：

```
请输入整数 x,y,z: 12,98,56
```