

本章导读：

前面学习了数据类型、常量和变量的定义,用运算符将常量和变量等连接起来构成了表达式,表达式能够描述和解决某项事务。那么用C程序怎么解决,也就是C语言如何编程实现?最基本的是能够进行输入和输出。设计自己的输入和输出格式是本章的重点:即格式化输入函数scanf和格式化输出函数printf。编程解决问题与解决现实问题相似,凡事不可能一帆风顺进行到底,总需要在某些时候进行选择,并对自己的选择负责。在C语言中,要理解结构化程序设计的三种基本结构,即顺序、选择和循环结构,在学习上要逐渐培养结构化程序设计的编程思想。

学习重点内容:

- (1) 理解三种程序控制结构的流程图。
- (2) 掌握数值型数据(整型和实型)的格式化输入输出方法。
- (3) 掌握字符型数据(char)的格式化输入输出方法。
- (4) 能够编写简单顺序控制的程序。

从程序流程的角度来看,程序可以分为三种基本结构,即顺序结构、选择(分支)结构、循环(重复)结构。这三种基本结构可以完成所有的各种复杂程序。C语言提供了多种语句来实现这些控制结构。

3.1 程序的控制结构

3.1.1 算法的基本概念

一个程序应包括:

(1) 对数据的描述。在程序中要指定数据的类型和数据的组织形式,即数据结构(data structure)。

(2) 对操作的描述。即操作步骤,也就是算法(algorithm)。

① 计算机算法:计算机能够执行的算法。

② 计算机算法可分为两大类。

- 数值运算算法:求解数值。
- 非数值运算算法:事务管理领域,查找、删除等。

Nikiklaus Wirth 提出的公式:

数据结构 + 算法 = 程序

教材认为：

程序 = 算法 + 数据结构 + 程序设计方法 + 语言工具和环境

这 4 个方面是一个程序设计人员所应具备的知识。

本课程的目的是使读者知道怎样编写一个 C 程序,进行编写程序的初步训练,因此,只介绍算法的初步知识。为后面章节的学习建立一定的基础。

3.1.2 算法的特性

- (1) 有穷性：一个算法应包含有限的操作步骤而不能是无限的。
- (2) 确定性：算法中每一个步骤应当是确定的,而不能是含糊的、模棱两可的。
- (3) 有效性：算法中每一个步骤应当能有效地执行,并得到确定的结果。
- (4) 有零个或多个输入。
- (5) 有一个或多个输出。

3.1.3 算法的表示

1. 用自然语言表示算法

除了很简单的问题,一般不用自然语言表示算法。

2. 用流程图表示算法

流程图表示算法,直观形象,易于理解,是比较常用的表示算法的方法。用流程图表示算法通常用图 3-1 所示的各种图形符号。

【例 3-1】 求 5! 的算法流程图表示(图 3-2 和图 3-3)。

【例 3-2】 判定闰年的算法用流程图 3-4 所示。

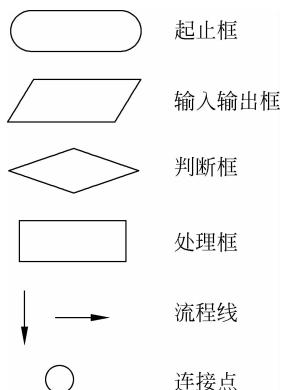


图 3-1 流程图符号

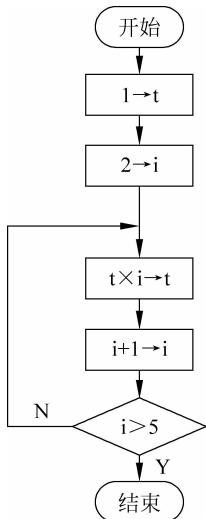


图 3-2 没有打印功能

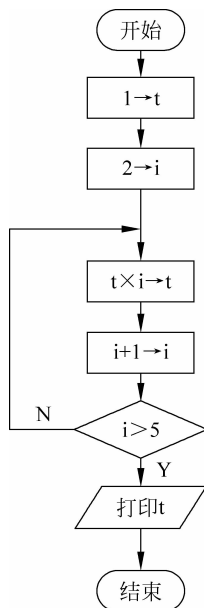


图 3-3 带有打印功能

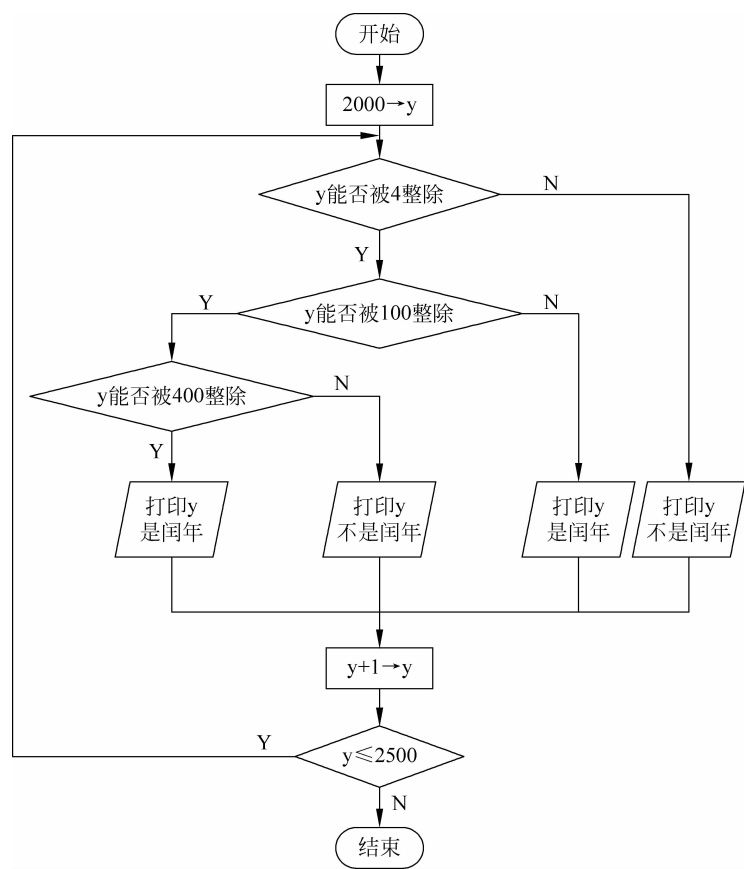


图 3-4 判断闰年流程图

3. 三种基本结构和改进的流程图

- (1) 顺序结构：如图 3-5 所示。
- (2) 选择结构：如图 3-6 所示。
- (3) 循环结构：如图 3-7 所示。

三种基本结构的共同特点：

- 只有一个入口；
- 只有一个出口；
- 结构内的每一部分都有机会被执行到；
- 结构内不存在“死循环”。

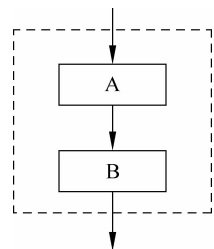


图 3-5 顺序结构

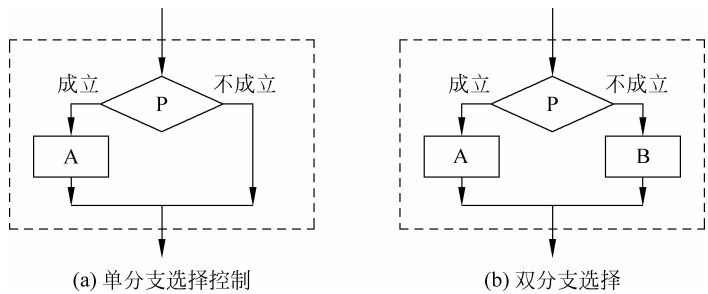


图 3-6 选择结构

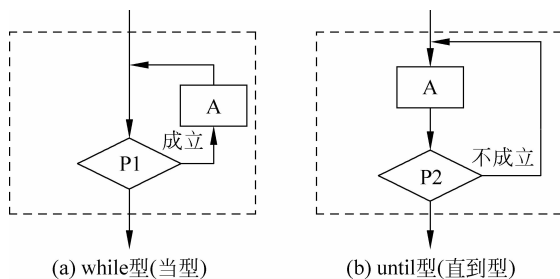


图 3-7 循环结构

3.1.4 用 N-S 流程图表示算法

1973 年美国学者提出了一种新型流程图：N-S 流程图。

- (1) 顺序结构：如图 3-8 所示。
- (2) 选择结构：如图 3-9 所示。
- (3) 循环结构：如图 3-10 所示。

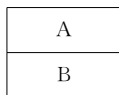


图 3-8 顺序结构的 N-S 图

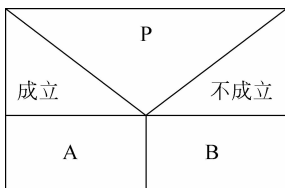


图 3-9 选择结构的 N-S 图

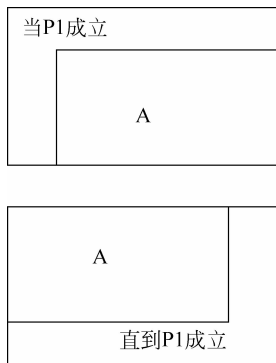


图 3-10 循环结构的 N-S 图

3.1.5 用计算机语言表示算法

我们的任务是用计算机解题，就是用计算机实现算法；用计算机语言表示算法必须严格遵循所用语言的语法规则。

【例 3-3】 用 C 语言求 5!。

```

1  #include <stdio.h>
2  void main()
3  {
4      int i,t;
5      t=1;
6      i=2;
7      while(i<=5)          //循环结构,判断一下循环条件是否为真
8      { t=t*i;

```

```
9      i = i + 1;
10   }
11   printf(" %d",t);
12 }
```

3.1.6 结构化程序设计方法

结构化程序设计所遵循的原则如下：

- 自顶向下；
- 逐步细化；
- 模块化设计；
- 结构化编码。

3.2 C 语句概述

C 程序的结构如图 3-11 所示。

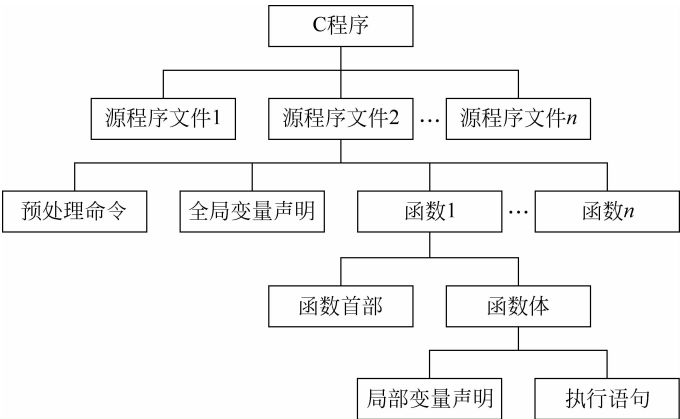


图 3-11 C 程序的结构图

C 程序的执行部分是由语句组成的。程序的功能也是由执行语句实现的。
C 语句可分为以下 5 类。

1. 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；

执行表达式语句就是计算表达式的值。

例如：

```
x = y + z;           //赋值语句
y + z;              //加法运算语句,但计算结果不能保留,无实际意义
```

```
i++; //自增 1 语句,i 值增加 1
```

2. 函数调用语句

由函数名、实际参数加上分号“;”组成。

其一般形式为：

```
函数名(实际参数表);
```

执行函数调用语句就是调用函数体并把实际参数赋予函数定义中的形式参数,然后执行被调函数体中的语句,求取函数值。

例如：

```
printf("C Program"); //调用库函数,输出字符串
```

3. 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 3 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：break 语句、goto 语句、continue 语句、return 语句。

4. 复合语句

把多个语句用括号{}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,而不是多条语句。

例如：

```
{ t = a;  
  a = b;  
  b = t;  
}
```

是一条复合语句。

复合语句内的各条语句都必须以分号“;”结尾,在括号“}”外不能加分号。

5. 空语句

只有分号“;”组成的语句称为空语句。空语句是什么也不执行的语句。在程序中空语句可用来作空循环体。

例如：

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

本语句的功能是：只要从键盘输入的字符不是回车则重新输入,这里的循环体为空语句。

3.3 数据输入输出的概念及在 C 语言中的实现

所谓输入输出是以计算机为主体而言的。本章介绍的是通过标准输入设备(键盘等)、标准输出设备(显示器等)完成数据输入和输出语句。在 C 语言中,所有数据的输入输出都是由库函数完成的,因此都是函数调用语句。在使用 C 语言库函数时,要用预编译命令 `#include` 将有关“头文件”包括到源文件中。使用标准输入输出库函数时要用到“`stdio.h`”文件,`stdio` 是 `standard input and outupt` 的意思。因此源文件开头应有以下预编译命令:

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

或

```
#include "stdio.h"
```

3.4 字符数据的输入与输出

3.4.1 字符输出函数 `putchar`

`putchar` 函数是字符输出函数,其功能是在显示器上输出单个字符。

其一般形式为:

```
putchar(字符型量)
```

例如:

```
putchar('A');           (输出大写字母 A)
putchar(x);             (输出字符变量 x 的值)
putchar('\101');        (也是输出字符 A)
putchar('\n');          (换行)
```

对控制字符则执行控制功能,不在屏幕上显示。

使用本函数前必须要用文件包含命令:

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

或

```
#include "stdio.h"
```

【例 3-4】 输出单个字符。

```
#include <stdio.h>
void main()
{
    char a = 'G', b = 'O', c = 'o', d = 'd';
    putchar(a); putchar(b); putchar(c); putchar(d);
    putchar('\t');
}
```

3.4.2 键盘输入函数 getchar

getchar 函数的功能是从键盘输入一个字符。

其一般形式为：

```
getchar();
```

通常把输入的字符赋予一个字符变量,构成赋值语句,例如:

```
char c;  
c = getchar();
```

【例 3-5】 输入单个字符。

```
#include <stdio.h>  
void main()  
{  
    char c;  
    printf("input a character\n");  
    c = getchar();  
    putchar(c);  
}
```

使用 getchar 函数还应注意以下 3 个问题:

- (1) 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。
- (2) 使用本函数前必须包含文件“stdio.h”。
- (3) 程序最后两行可用下面两行的任意一行代替:

```
putchar(getchar());  
printf(" %c",getchar());
```

3.5 格式化输出函数 printf

在 C 语言中,printf 函数是主要的数据(常量、变量、表达式等)输出函数,可以替代所有的输出函数。其关键字最末一个字母 f 即为“格式”(format)之意。其功能是按用户指定的格式,把指定的数据显示到显示器屏幕上。在前面的例题中已多次使用过这个函数。

3.5.1 printf 函数调用的一般形式

printf 函数是一个标准库函数,它的函数原型在头文件“stdio.h”中。

printf 函数调用的一般形式为:

```
printf("格式控制字符串",输出表列)
```


其中：格式控制字符串用于指定输出格式。格式控制字符串可由格式字符串和非格式字符串两种组成。格式字符串是以 % 开头的字符串，在 % 后面跟有各种格式字符，以说明输出数据的类型、形式、长度、小数位数等。如：“%d”表示按十进制整型输出；“%c”表示按字符型输出等。

非格式字符串在输出时原样输出，在显示中起提示作用。

输出表列中给出了各个输出项，要求格式字符串的控制字符和各输出项在个数上、类型上和顺序上要一一对应。

在 C 语言中格式字符串的一般形式为：

%[标志][输出最小宽度][.精度][长度]格式字符

其中方括号[]中的项为可选项。

各项的意义介绍如下。

(1) 格式字符：格式字符用以表示输出数据的类型，其格式符和意义如表 3-1 所示。

表 3-1 输出数据时格式字符及其意义

格式字符	意 义
d	以十进制形式输出带符号整数(正数不输出符号)
o	以八进制形式输出无符号整数(不输出前缀 0)
x, X	以十六进制形式输出无符号整数(不输出前缀 0x)
u	以十进制形式输出无符号整数
f	以小数形式输出单、双精度实数
e, E	以指数形式输出单、双精度实数
g, G	以%f或%e中较短的输出宽度输出单、双精度实数
c	输出单个字符
s	输出字符串

(2) 标志：标志字符为一、+、0、空格 4 种，其意义如表 3-2 所示。

表 3-2 标志字符及意义

标志	意 义
—	结果左对齐，右边填充空格
+	输出符号(正号或负号)
空格	输出值为正时冠以空格，为负时冠以负号
0	右对齐时，如果实际宽度小于 width，则在左边的空位上补 0

(3) 输出最小宽度：用十进制整数来表示输出的最少位数。若实际位数多于定义的宽度，则按实际位数输出，保证不会因为输出而使数据发生截断误差。若实际位数少于定义的宽度则按表 3-2 补以空格或 0。

(4) 精度：精度格式符以“.”开头，后跟十进制整数。本项的意义是：如果输出的是数值，则表示小数的位数；如果输出的是字符串，则表示输出字符的个数；若实际位数大于所定义的精度数，输出数值则四舍五入截去超过的部分。

- 长度：长度格式符为 h,l 两种,h 表示按短整型(short)量输出,l 表示按长整型(long)量输出。

【例 3-6】 输出整数。

```
1 #include <stdio.h>
2 void main()
3 {   int a = 88888, b = 89;
4     printf(" %4d %4o\n", a, b);
5     printf(" % +4hd, % -4d\n", a, b);
6     printf(" % +04d, % -04d\n", a, b);
7     printf("a= %d, b= %d\n", a, b);
8 }
```

执行结果：

```
88888 131
+23352.89
+88888.89
a=88888,b=89
Press any key to continue
```

本例中 4 次输出了 a,b 的值,但由于格式控制串不同,输出的结果也不相同。第 4 行的输出语句格式控制串中,两格式串 %4d %4o 之间加了一个空格(非格式字符),所以输出的 a,b 值之间有一个空格,a 的宽度大于指定宽度原样输出,b 的值输出右对齐,左补空格,占 4 列宽。第 5 行的 printf 语句格式控制串中加入的是标志字符、长度字符 h 和非格式字符逗号,因此输出的 a,b 值之间加了一个逗号,%+4hd 输出带“+”的右对齐的 short 整型数据,由于 88888 超出了 short 型数据的取值范围,宽度大于指定宽度,所以按实际宽度输出,数值取 88888 的低位 2 字节即为 23352;%-4d 左对齐占 4 列宽。第 6 行的格式控制字符串 % +04d 要求按右对齐左补“0”、带“+”占 4 列宽输出 a 的值。%-04d 要求按左对齐占 4 列宽输出 b 的值。第 7 行中为了提示输出结果又增加了非格式字符串,非格式字符串原样输出。

【例 3-7】 按格式控制字符串要求输出浮点型小数和字符型数据。

```
1 #include <stdio.h>
2 void main()
3 {
4     int a = 15;
5     float b = 123.1234567f;
6     double c = 12345678.1234567;
7     char d = 'p';
8     printf("a = %d, %5d, %o, %X\n", a, a, a, a);
9     printf("b = %f, %lf, %5.4lf, %e\n", b, b, b, b);
10    printf("c = %lf, %f, %8.4lf\n", c, c, c);
11    printf("d = %c, %8c\n", d, d);
12 }
```

执行结果：

```
a=15, 15,17,F
b=123.123459,123.123459,123.1235,1.231235e+002
c=12345678.123457,12345678.123457,12345678.1235
d=p, p
Press any key to continue
```

本例第 8 行中以 4 种格式输出整型变量 a 的值,其中“%5d”要求输出宽度为 5,右对齐左补空格,而 a 值为 15 只有两位故补三个空格。第 9 行中以 4 种格式输出实型量 b 的值。其中“%f”和“%lf”格式的输出生相同,说明“l”符对“f”类型无影响。“%5.4lf”指定输出宽度为 5,精度为 4,由于实际长度超过 5 故应该按实际位数输出,小数位数超过 4 位部分被截去。第 10 行输出双精度实数,“%8.4lf”由于指定精度为 4 位故截去了超过 4 位的部分。第 11 行输出字符变量 d,其中“%8c”指定输出宽度为 8,故在输出字符 p 之前补加 7 个空格。

3.5.2 使用 printf 函数的注意事项

除此之外,使用 printf 函数时还应补充以下几方面。

(1) 如果想输出%,则应该在“格式控制”字符串中用连续两个%表示。例如:

```
printf("%f% %",1.0/3);
```

则输出: 0.333333%

(2) [输出最小宽度][.精度]仅用于输出浮点型小数和字符串时的格式控制。

- 输出浮点型小数: 形如 %[—]m.nf,f 是一个浮点型变量,当按此格式输出时,包括小数点共 m 列宽(按照对齐方式补齐)、取 n 位小数;当 f 的实际宽度大于 m 时,则按实际宽度输出。

例如:

```
float f = -3.14587f;
printf("% -10.2f",f);      //输出数据占 10 列宽,取 2 位小数,左对齐
printf("% 5.2f",f);        //输出数据占 5 列宽,取 2 位小数
printf("% 8.2f",f);        //输出数据占 8 列宽,取 2 位小数,右对齐
printf("% 3.2f\n",f);      //输出数据占 3 列宽,取 2 位小数
                           //f 的实际宽度大于 3 时,按实际宽度输出
```

则执行结果为: -3.15 _ _ _ _ _ -3.15 _ _ _ -3.15 -3.15

- 输出字符串: 形如 %[—]m.ns,s 是一个字符串,当按此格式输出时,按照对齐方式用空格补齐 m 列、取 n 个字符;当 s 的实际宽度大于 m 时,则按实际宽度输出。

例如:

```
printf("% 3s, % 7.2s, % .4s, % - 5.3s\n","CHINA","CHINA","CHINA","CHINA");
```

输出为: CHINA,_ _ _ _ _ CH,CHIN,CHI

(3) printf() 函数格式控制字符串可以包含转移字符。

3.6 格式化输入函数 scanf

scanf 函数称为格式输入函数,即按用户指定的格式从键盘上把数据输入到指定的变量之中。

3.6.1 scanf 函数调用的一般形式

scanf 函数是一个标准库函数,它的函数原型在头文件“stdio.h”中,在使用 scanf 函数之前要包含 stdio.h 文件。

scanf 函数的一般形式为:

```
scanf("格式控制字符串",地址表列);
```

其中,格式控制字符串的作用与 printf 函数相同,但不能显示非格式字符串,也就是不能显示提示字符串。地址表列中给出各变量的地址。地址是由地址运算符“&”后跟变量名组成的。

例如:

```
&a, &b
```

分别表示变量 a 和变量 b 的地址。

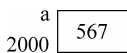
这个地址是编译系统在内存中给 a,b 变量分配的地址。在 C 语言中,使用了地址这个概念,这是与其他语言不同的。应该把变量的值和变量的地址这两个不同的概念区别开来。变量的地址是 C 编译系统分配的,用户不必关心具体的地址是多少。

变量的地址和变量值的关系如下:

在赋值表达式中给变量赋值,例如:

```
a = 567;
```

则 a 为变量名,567 是变量的值,&a 是变量 a 的地址(假设地址为 2000,即 &a=2000)。图示为:



但在赋值号左边是变量名,不能写地址,而 scanf 函数在本质上也是给变量赋值,但要求写变量的地址,如 &a。这两者在形式上是不同的。& 是一个取地址运算符,&a 是一个表达式,其功能是求变量的地址。

【例 3-8】 scanf 函数的使用。

```
1  #include <stdio.h>
2  void main()
3  {
4      int a,b,c;
5      printf("input a,b,c\n");
6      scanf(" %d %d %d",&a,&b,&c);
7      printf("a = %d,b = %d,c = %d",a,b,c);
8  }
```

在本例中,由于 scanf 函数本身不能显示提示串,故先用 printf 语句在屏幕上输出提示,请用户输入 a、b、c 的值。执行 scanf 语句,假如用户输入 7 8 9 后按 Enter 键,则 3 个数被提交给相应的内存地址空间,即 a,b,c 的内存地址空间。在 scanf 语句的格式串中由于没有非格式字符在"%d%d%d"之间作输入时的间隔,因此在输入时要用一个或一个以上的空格、回车键、Tab 键作为每两个输入数之间的间隔。例如:

```
7 _8 _9 ↵
```

或

- 7 ✓
- 8 ✓
- 9 ✓

格式字符串的一般形式为：

```
%[*][输入数据宽度][长度]格式字符
```

其中有方括号[]的项为任选项，各项的意义如下：

(1) 格式字符：表示输入数据的类型，其格式符和意义如表 3-3 所示。

表 3-3 输入数据时格式字符及意义

格 式 字 符	字 符 意 义
d	输入十进制整数
o	输入八进制整数
x,X	输入十六进制整数
u	输入无符号十进制整数
f 或 e	输入实型数(用小数形式或指数形式)
c	输入单个字符
s	输入字符串

(2) “*”：抑制符，用以表示输入的数据不赋值给相应的变量，即跳过该输入值。

例如：

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

当输入为 1 2 3 时，把 1 赋予 a，2 被跳过，3 赋予 b。

(3) 宽度：用十进制整数指定输入的宽度（即字符数），遇空格或不可转换字符则结束。

例如：

```
scanf("%5d",&a);
```

输入：12345678

只把 12345 赋予变量 a，其余部分被截去。

又如：

```
scanf("%4c %2c",&c1,&c2);
```

输入：abcdefgh ✓

将把 abcd 中的 a 赋予 c1，而把 ef 中的 e 赋予 c2。

主要用于从文件读取序列数据，根据域宽依次赋给各个变量。

(4) 长度：长度格式符为 l 和 h，%hf 表示输入短整型数据，%lf 表示输入双精度浮点数。

3.6.2 使用 scanf 函数的注意事项

(1) scanf 函数中没有精度控制，如 scanf("%5.2f",&a);是非法的。不能企图用此语

句输入小数为 2 位的实数,不能有转义控制字符。

(2) scanf 中要求给出变量地址,如给出变量名则会出错。如 scanf("%d",a);是非法的,应改为 scanf("%d",&a);才是合法的。

(3) 在输入多个数值数据时,若格式控制串中没有非格式字符作输入数据之间的间隔则可用空格、Tab 或回车符作间隔。C 编译在碰到空格、Tab、回车或非法数据(如对"%d"输入"12A"时,A 即为非法数据)时即认为该数据结束。

(4) 在输入字符数据时,特别需要注意空格符、回车符均作为有效字符使用。若格式控制串中无非格式字符,则认为所有输入的字符均为有效字符。

例如:

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

输入为:

```
d_e_f
```

则把'd'赋予 a,'_'赋予 b,'e'赋予 c。

只有当输入为:

```
def
```

时,才能把'd'赋予 a,'e'赋予 b,'f'赋予 c。

如果在格式控制中加入空格作为间隔,例如:

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

则输入时各数据之间可加空格。

注意: 同时输入数值和字符时,可以用空格吸收回车符。

例如:

```
int a,b;
```

```
char c;
```

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

```
printf("a = %d,b = %d,c = %d",a,b,c);
```

若输入 23 45 a ↵, 则 a = 23, b = 45, c = 32(空格的 ASCII 为 32);

若输入 23 45a ↵, 则 a = 23, b = 45, c = 97

```
int a,b;
```

```
char c;
```

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

```
printf("a = %d,b = %d,c = %d",a,b,c);
```

若输入 23 ↵

45 ↵

a ↵

则 a = 23, b = 45, c = a(空格的 ASCII 为 32); 空格吸收了回车符。

(5) 如果格式控制串中有非格式字符则输入时也要原样输入该非格式字符。

例如:

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

其中用非格式符“,”作间隔符,故输入时应为:

5,6,7

又如:

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

则输入应为:

a = 5,b = 6,c = 7

【例 3-9】 数据的格式化输入输出。输入一学生的学号(8 位数字)、生日(年-月-日)、性别(M: 男,F: 女)及三门功课(语文、数学、英语)的成绩,现要求计算该学生的总分和平均分,并将该学生的全部信息输出(包括总分、平均分)。

```
#include <stdio.h>
void main()
{
    unsigned long no;           //学号
    unsigned int year, month, day; //生日
    unsigned char sex;          //性别
    float chinese, math, english;
    float total, average;
    printf("input the student's NO:");
    scanf("%8ld",&no);
    printf("input the student's Birthday(yyyy-mm-dd): ");
    scanf("%4d-%2d-%2d",&year,&month,&day);
    fflush(stdin);              //清除键盘缓冲区
    printf("input the student's Sex(M/F)");
    scanf("%c",&sex);
    printf("input the student's Score(chinese,math,english): ");
    scanf("%f%f%f",&chinese,&math,&english);
    total = chinese + math + english;
    average = total/3;
    printf("\n=== No ===== birthday == sex == chinese == math == english ==\n");
    printf("%8ld %4d-%2d-%2d %c %f - 5.1f %f - 5.1f %f - 5.1f\n", no, year, month, day, sex, chinese, math, english, total, average);
}
```

执行结果:

```
input the student's NO:20130401
input the student's Birthday(yyyy-mm-dd): 1993-9-6
input the student's Sex(M/F)F
input the student's Score(chinese,math,english): 95.89.94

===No=====birthday==sex==chinese==math==english==total==average
20130401 1993-09-06 F 95.0 89.0 94.0 278.0 92.7
```

3.7 顺序结构程序设计举例

【例 3-10】 输入三角形的三边长,求三角形的面积。

已知三角形的三边长 a, b, c , 则该三角形的面积公式为:

$$\text{area} = \sqrt{s(s-a)(s-b)(s-c)}$$

其中 $s = (a+b+c)/2$ 。

源程序如下:

```
#include <stdio.h>
#include <math.h>
void main()
{
    float a, b, c, s, area;
    scanf("%f, %f, %f", &a, &b, &c);
    s = 1.0/2 * (a + b + c);
    area = sqrt(s * (s - a) * (s - b) * (s - c)); //sqrt 函数在 math.h 中声明
    printf("a = %7.2f, b = %7.2f, c = %7.2f, s = %7.2f\n", a, b, c, s);
    printf("area = %7.2f\n", area);
}
```

【例 3-11】 求 $ax^2+bx+c=0$ 方程的根, a, b, c 由键盘输入, 设 $b^2-4ac>0$ 。

求根公式为:

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}, \quad \text{令 } p = \frac{-b}{2a}, q = \frac{\sqrt{b^2 - 4ac}}{2a}$$

则 $x_1 = p + q$

$x_2 = p - q$

源程序如下:

```
#include <stdio.h>
#include <math.h>
void main()
{
    float a, b, c, disc, x1, x2, p, q;
    scanf("a = %f, b = %f, c = %f", &a, &b, &c);
    disc = b * b - 4 * a * c;
    p = -b / (2 * a);
    q = sqrt(disc) / (2 * a);
    x1 = p + q; x2 = p - q;
    printf("\nx1 = %5.2f\nx2 = %5.2f\n", x1, x2);
}
```

【例 3-12】 从键盘输入一个大写字母, 要求改用小写字母输出。英文字母大小写相对的 ASCII 码值相差 32, 例如 'a' 比 'A' 的 ASCII 值大 32, 以此类推。

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



```
{
    char c1,c2;
    c1 = getchar();
    printf(" %c, %d\n",c1,c1);
    c2 = c1 + 32;
    printf(" %c, %d\n",c2,c2);
}
```

执行结果：

```
A
A,65
a,97
Press any key to continue
```

本章小结

(1) 对于输入输出函数的参数,格式控制字符串基本相同,主要区别是 printf 可以指定精度和包含转义字符,scanf 则不能;再就是 printf 后面参数是输出项列表;而 scanf 后面参数是地址表列。

(2) 对于输入输出函数形式,格式控制字符串包含格式符和非格式符。对于 printf 非格式符原样输出;而对于 scanf 如果有非格式符则在进行输入操作时必须按照原样格式输入。

(3) 注意有“%c”的 scanf 函数的使用。

习 题 3

1. C 语言中的语句有哪几类? 它们是什么?
2. C 语言中表达式和表达式语句的区别是什么? 什么时候用表达式,什么时候用表达式语句?
3. 请写出下面程序的输出结果:

```
#include <stdio.h>
void main()
{
    int a = 5, b = 7;
    float x = 67.8564, y = -123.456;
    char c = 'A';
    printf(" %d %d\n", a, b);
    printf(" %3d %3d\n", a, b);
    printf(" %f, %f\n", x, y);
    printf(" % -10f. %8.2f, % .4f, %3f\n", x, y, x, y);
    printf(" %s, %5.2s\n", "CHINA", "CHINA");
}
```

4. 按照 printf 函数格式控制字符串的使用规则,写出 printf 语句,要求按照给定格式输出。

假设 `int a=4,b=5; float x=2.3,y=-4.5; c1='a',c2='b';`

要求如下:

```
a = 4    b = 5
x = 2.300000, y = -4.500000
x + y = -2.20    x - y = 6.80
c1 = 'a' or 97(ASCII)
c2 = 'b' or 98(ASCII)
```

5. 用下面的 `scanf` 函数输入数据,使 `a=4,b=5,x=2.3,y=-4.5,c1='a',c2='b'`,那么在键盘上如何输入?

```
#include <stdio.h>
void main()
{
    int a,b;
    float x,y;
    char c1,c2;
    scanf("a = %d b = %d",&a,&b);
    scanf("_ %f _ %f",&x,&y);
    scanf("_ %c _ %c",&c1,&c2);
}
```

6. 设圆半径用 `r` 表示,圆柱高用 `h` 表示,求圆周长、圆面积、圆球表面积、圆球体积、圆柱体积。用 `scanf` 输入数据,`printf` 输出计算结果,要求有适当的提示,取小数点后两位数字,请编程。

7. 输入一个华氏温度,要求输出摄氏温度。公式为

$$c = \frac{5}{9}(F - 32)$$

输出要有说明,取两位小数,请编程。

8. 用 `getchar` 函数读入两个字符给 `c1,c2`,然后分别用 `putchar` 函数和 `printf` 函数输出这两个字符,请编程。