

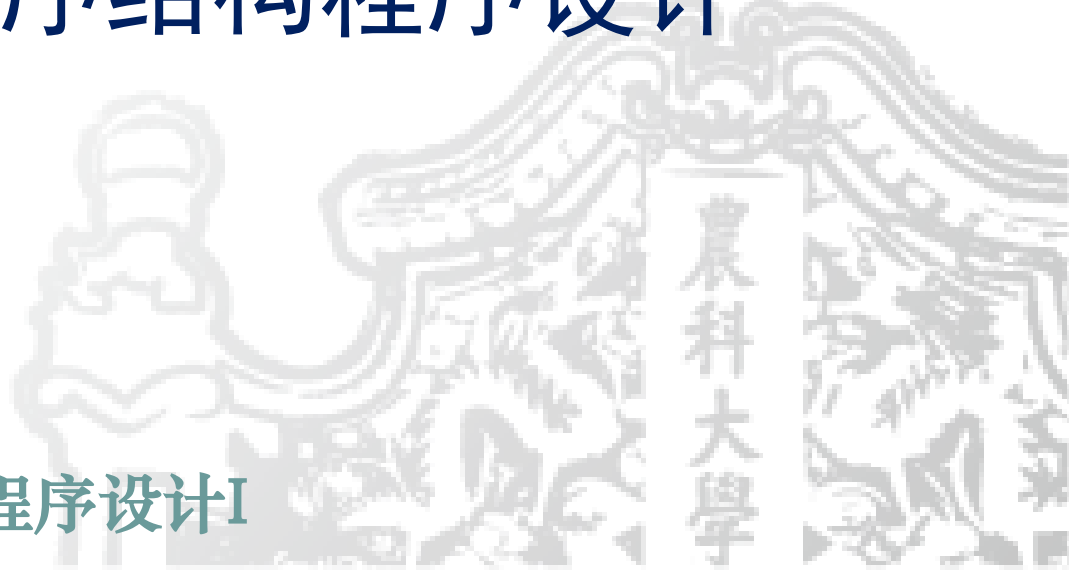


信息与电气工程学院

C程序设计案例教程

第4章 顺序结构程序设计

程序设计I





本章主要内容有

- 顺序结构程序流程控制
- C语句概述
- 赋值语句
- 字符与字符串输入函数
- 字符与字符串输出函数
- 格式化输入与输出函数应用



4.1 顺序结构程序流程控制

顺序结构是按顺序执行各个命令或程序模块的操作，即A命令的操作与B命令的操作是顺序执行的关系，如图4.2所示。

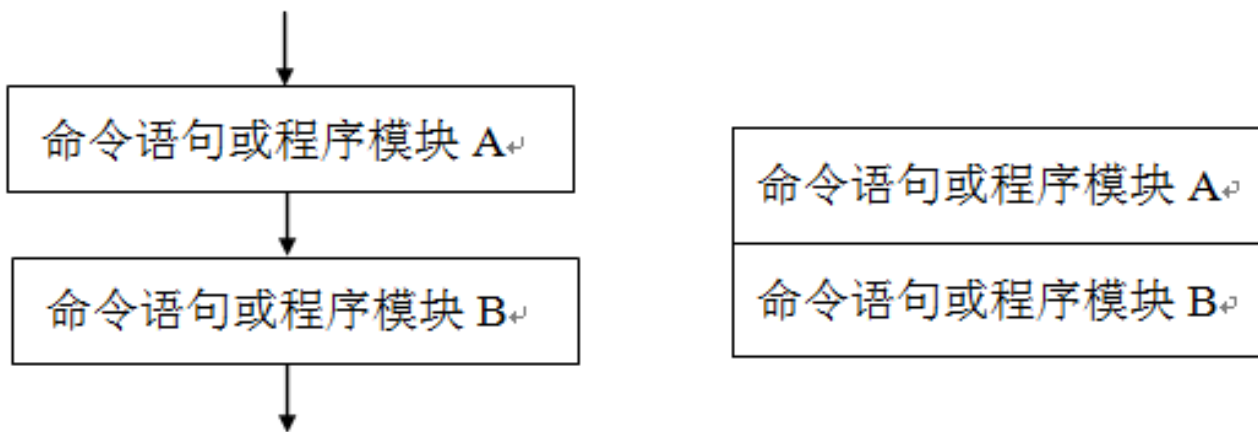


图4.1顺序结构程序流程图与N-S图



4.2 C语句概述

★C语句分为5类

❖控制语句：共9种

- 完成一定的控制功能，这些语句能够根据一定的测试条件决定某些语句是否被执行，如分支、循环、跳转等语句。

if()~else~	(条件语句)
for()~	} (循环语句)
while()~	
do~while()	
continue	(结束本次循环语句)
switch	(多分支选择语句)
break	(终止switch或循环语句)
goto	(转向语句)
return	(从函数返回语句)



- ❖ 函数调用语句：由函数加“;”组成

```
printf("This is a C program");
```

- ❖ 表达式语句：由表达式加“;”组成

- 典型的是赋值语句，由赋值表达式加分号构成，如： $y=x+1$ 是表达式，而 $y=x+1;$ 则是赋值语句。实际上函数调用语句也属于表达式语句。

```
a=3 ;  
i++ ;
```

- ❖ 空语句：只有一个“;”，什么也不做

```
for(i=0;i<100;i++)  
{  
    ;  
}
```



❖ 复合语句:

- 用 {...}括起来的一组语句, 也称作程序块。
- 一般形式:

```
int i=0,sum=0;
while(i<=100)
{
    sum=sum+i;
    i=i+1;
}
```

```
int i=0,sum=0;
while(i<=100)
    sum=sum+i;
    i=i+1;
```

● 说明

- ◆ 复合语句“}”后不加分号
- ◆ 语法上和其它语句相同
- ◆ 复合语句可嵌套

有什么问题?



4.3 赋值语句

❖ 基本格式:

赋值表达式 = ;

- 赋值表达式可以包含在其它表达式中，而赋值语句不可以。

如果t的初始值为7

a=3,b=8的情况下,t=?

a=3,b=0的情况下,t=?

a=0,b=0的情况下,t=?

例: `if ((a=b)>0) t=a;`

不可写为: `if ((a=b;)>0) t=a;`



❖ 连续赋值语句

```
int a , b , c ;  
a=b=c=1 ;
```



```
int a=b=c=1 ;
```



步骤： 连续赋值语句应“**从右向左**”计算

```
a=b=c=1 ;
```

等价于：

```
a=(b=(c=1)) ;
```

1. 把常数 1 赋给变量c，表达式(c=1)的值为1；
2. 把表达式(c=1)赋给变量b，表达式(b=(c=1))的值为1；
3. 将右侧表达式的值1赋给a，a = 1。



4.4 数据输入输出

- ❖ 所谓数据的输入输出是从计算机角度出发的。
- ❖ C语言本身没有I/O语句，所有I/O都由函数来实现。
- ❖ C语言的标准库函数
 - 由编译系统提供的一系列函数，以库形式存放在系统中，不是C语言文本的组成部分。
 - 库函数已编译成目标文件（.obj），在连接阶段才与源程序编译成的目标文件相连接，生成可执行文件。
 - 调用形式：
函数名（参数表）
 - 注意在调用C语言库函数时，需要使用编译预处理命令#include <相关的头文件>，使相应的头文件包含到用户源程序中。

常用头文件：

stdio.h 定义输入输出函数

string.h 定义字符串操作函数

math.h 定义sin、cos等数学函数

❖ 标准输入输出函数

putchar 输出字符， scanf 格式输入， puts 输出字符串

getchar 输入字符， printf 格式输出， gets 输入字符串

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

```
或 #include "stdio.h"
```



4.5 字符数据的输入输出

★ putchar 函数（单字符输出函数）

格式：`putchar(‘字符’);`
或 `putchar(字符变量);`

强调：被输出的单个字符必须被 ‘ ’ 括起来

❖ 输出一个字符：

运行结果：BOY



结果当然还是一样的！

```
/* 例1 putchar(字符变量) */  
#include <stdio.h>  
int main()  
{ char a,b,c;  
  a='B'; b='O'; c='Y';  
  putchar(a); putchar(b); putchar(c);  
}
```



❖ 输出控制字符

```
/* 例2 putchar(‘控制字符’) */  
#include <stdio.h>  
int main()  
{ char a,b;  
  a='O'; b='k';  
  putchar(a); putchar('\n'); putchar(b);  
}
```

运行结果: O
k

❖ 输出转义字符

运行结果: AB

```
/*例3 putchar() 含有转义符*/  
#include <stdio.h>  
int main()  
{ char a;  
  a='B';  
  putchar('\101'); putchar(a);  
}
```



★ getchar 函数（单字符输入函数）。

格式：`getchar()`；

强调：输入单个字符后，**必须按一次回车**，计算机才接受输入的字符。

```
/*例4 */  
#include <stdio.h>  
int main()  
{ char c;  
  c=getchar();  
  putchar(c);  
}
```

运行结果：

```
a↵  
a
```



include <stdio.h>不能少！



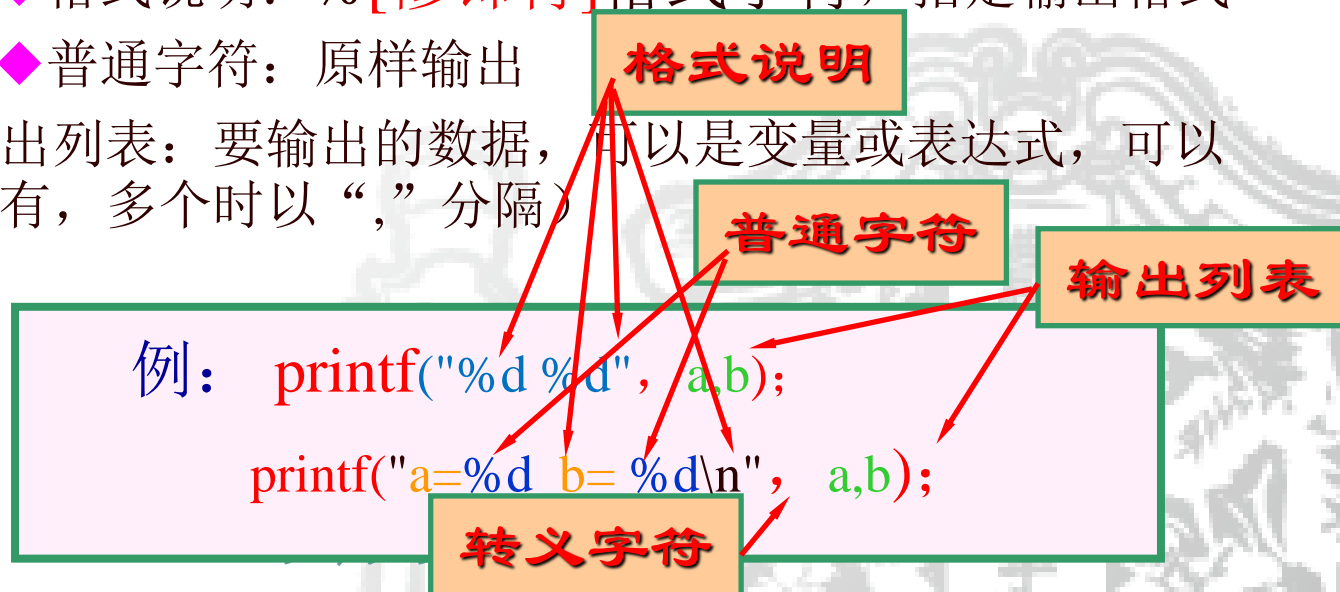
4.6 格式输入与输出

★printf函数（格式输出函数）

❖printf函数的一般格式

格式：`printf(格式控制, 输出列表);`

- 格式控制：用双引号括起来的字符串，包含两种信息
 - ◆ 格式说明：`%[修饰符]`格式字符，指定输出格式
 - ◆ 普通字符：原样输出
- 输出列表：要输出的数据，可以是变量或表达式，可以没有，多个时以“,”分隔





信息与电气工程学院

```
/*例5 从最简单的学起 */  
#include <stdio.h>  
int main()  
{  
    printf("Hello World !" );  
}
```

运行结果：
Hello World!





★ 格式字符

❖ d格式符：输出十进制整数，有3种用法

● %d格式：按数据实际长度输出。

```
/* 例6 %d 格式的 printf() 函数使用 */  
#include <stdio.h>  
int main()  
{  
    int a=3 , b=4;  
    printf("%d %d\n ",a,b);  
    printf("a=%d , b=%d\n",a,b);  
}
```

运行结果：

3 4

a=3, b=4



格式说明决定最终输出的格式

格式说明应与输出列表项个数相同，顺序一致

格式说明通常用小写字母表示



- %md格式：m指定输出字段的宽度
 - ◆ 数据位数小于m，左端补空格，反之按实际输出。

```
int a=123 , b=12345 ;  
printf("%4d %4d", a , b);
```

```
 123 12345
```

- %ld格式：输出长整型数据
 - ◆ 可以用%mld格式指定输出列宽

```
long c=135790  
printf("%ld \n", c);  
printf("%8ld ", c);
```

```
135790  
 135790
```




❖ o格式符：八进制输出整数

- 是将内存中的二进制位整个按八进制输出，所以输出值没有符号。
- 可以指定输出宽度%mo，长整型可以用%lo格式输出。

❖ x格式符：十六进制输出整数

- 同o格式符，无符号，即无负十六进制数。
- 可以指定输出宽度%mx，长整型可以用%lx格式输出。

```
int a= -1;  
printf("%d, %o, %8o, %x", a, a, a, a);
```

输出:

-1, 177777, __177777, ffff

11	11	11	11	11	11	11	11
----	----	----	----	----	----	----	----

-1在内存的存放形式（补码）



❖ u格式符：十进制输出unsigned型数据

- int型可以用%u格式输出，unsigned型也可以用%d、%o和%x格式输出。

例7 无符号数据的输出

```
#include <stdio.h>
int main()
{ unsigned int a=65535;
  int b= -2;
  printf("a=%d , %o , %x , %u\n",a,a,a,a);
  printf("b=%d , %o , %x , %u\n",b,b,b,b);
}
```

运行结果：

```
a= - 1 , 177777 , ffff , 65535
b= - 2 , 177776 , fffe , 65534
```



❖ c格式符：输出一个字符

- 值在0~255的整数，可以用%c形式输出为字符

例8 字符数据的输出

```
#include <stdio.h>
int main()
{ char c='a';
  int i=97;
  printf("%c , %d\n",c,c);
  printf("%c , %d\n",i,i);
}
```

运行结果：

a , 97

a , 97



❖ s格式符：输出一个字符串

- 有%s, %ms, %-ms, %m.ns, %-m.ns五种用法

%s: 例如:printf("%s", "CHINA")输出"CHINA"字符串（不包括双引号）。

%ms: 输出的字符串占m列，如字符串本身长度大于m，则突破获m的限制,将字符串全部输出。若串长小于m，则左补空格。

%-ms: 如果串长小于m，则在m列范围内，字符串向左靠，右补空格。

%m.ns: 输出占m列，但只取字符串中左端n个字符。这n个字符输出在m列的右侧，左补空格。

%-m.ns: 其中m、n含义同上，n个字符输出在m列范围的左侧，右补空格。如果n>m，则自动取n值，即保证n个字符正常输出。



例9 字符串的输出

```
#include <stdio.h>
int main()
{
    printf("%3s , %7.2s , %.4s , %-5.3s\n", "CHINA",
        "CHINA", "CHINA", "CHINA");
}
```

m自动=n

运行结果:

CHINA , _____CH , CHIN , CHI _ _



❖ f格式符：输出实数

- %f格式：整数部分全部输出，小数6位。可以有非有效数字输出，因为单精度有效位7位，双精度16位。
- %m.nf格式：占m列，其中n位小数，左补空格。
- %-m.nf格式：右补空格

例10 %f格式输出实数时指定小数位数

```
#include <stdio.h>
int main()
{ float f=123.456001;
  printf("%f %10f %10.2f %.2f %-10.2f\n",f,f,f,f,f);
}
```

运行结果：

```
123.456001__123.456001____123.46__123.46__123.46____
```



❖ e格式符：指数形式输出实数

- %e格式：不指定m和n，小数6位，指数部分共5位，其中e和指数符号各1位，指数值3位。
- %m.ne和% -m.ne格式：m、n、-的含义同前面。
没有n时，自动=6

❖ g格式符：输出实数

- 可以自动根据数值大小选择 f 或 e 格式（选列少的）
- 不输出无意义的零





❖ 格式字符表

d	十进制整数	<code>int a=100;printf ("%d",a);</code>	100
x,X	十六进制无符号整数	<code>int a=255;printf("%x",a);</code>	ff
o	八进制无符号整数	<code>int a=8;printf("%o",a);</code>	10
u	不带符号十进制整数	<code>int a=100;printf("%u",a);</code>	100
c	单一字符	<code>char a=65;printf("%c",a);</code>	A
s	字符串	<code>printf("%s", "ABC");</code>	ABC
e,E	指数形式浮点小数	<code>float a=567.789;printf("%e",a);</code>	5.677890e+002
f	小数形式浮点小数	<code>float a=567.789;printf("%f",a);</code>	567.789000
g,G	e和f中较短一种	<code>float a=567.789;printf("%g",a);</code>	567.789
%%	百分号本身	<code>printf("%%");</code>	%



★scanf 函数 (格式输入函数)

❖一般形式

格式: `scanf(格式控制, 地址列表);`

- 功能: 按指定格式从键盘读入数据, 存入地址表指定的存储单元中, 并按回车键结束
- 格式控制: 含义同printf函数
- 地址列表: 变量地址或字符串地址, 地址间“,”分隔。
- 强调: 地址列表中每一项必须以取地址运算符&开头。

输入: 3_4_5↵
输出: 3,4,5

例11 用scanf函数输入数据

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

```
int main()
```

```
{ int a , b , c ;
```

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

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

```
}
```



例 `scanf("%4d%2d%2d",&yy,&mm,&dd);`

输入 19991015 ↵

则 1999 \Rightarrow yy, 10 \Rightarrow mm, 15 \Rightarrow dd

例 `scanf("%3d%*4d%f",&k,&f);`

输入 12345678765.43↵

则 123 \Rightarrow k, 8765.43 \Rightarrow f

例 `scanf("%2d%*3d%2d",&a,&b);`

输入 1234567↵

则 12 \Rightarrow a, 67 \Rightarrow b



❖ 输入分隔符的指定

- 一般以空格、TAB或回车键作为分隔符
- 输入数据时，遇非法输入则认为数据结束
- 其它字符做分隔符：格式串中两个格式符间有其它字符，则输入时对应位置也要有相同的字符。

例 `scanf("%d:%d:%d",&h,&m,&s);`
输入 12:30:45↵
则 12 \Rightarrow h, 30 \Rightarrow m, 45 \Rightarrow s

例 `scanf("%d%c%f",&a,&b,&c);`
输入 1234a123o.26 ↵
则 1234 \Rightarrow a, 'a' \Rightarrow b, 123 \Rightarrow c



例 `scanf("%c%c%c",&c1,&c2,&c3);`

输入 a b c ↵

则 $a \Rightarrow c1, \Rightarrow c2, b \Rightarrow c3$

