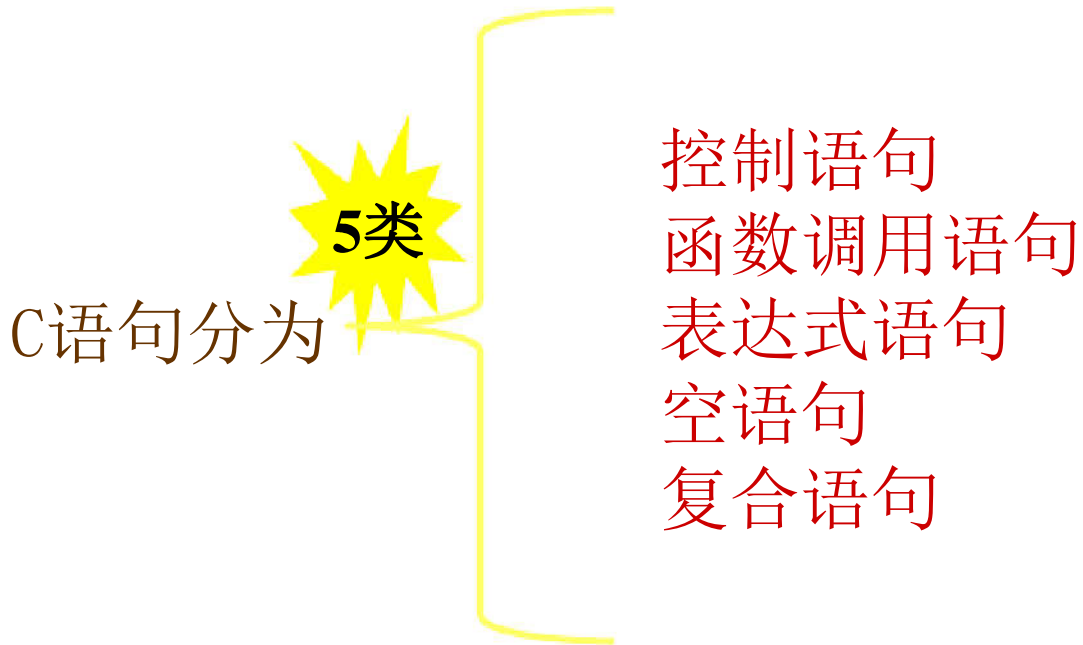


第四章

最简单的C程序设计

顺序结构

4.1 C语句概述



4.1 C语句概述

(一) 控制语句

完成一定的控制功能

1 **if() ~else** 条件语句

6 **break** 间断语句

2 **for()~** 循环语句

7 **switch()** 开关语句

3 **while()~**循环语句

8 **goto** 转向语句

4 **do ~while();**循环语句

9 **return** 返回语句

5 **continue** 继续语句

4.1 C语句概述

(二) 函数调用语句

有一个函数调用加一个分号构成一个语句



```
Printf("This is a C statement.");
```

4.1 C语句概述

(三) 表达式语句

有一个表达式加一个分号构成一个语句

例:

赋值表达式

+

分号

=

表达式语句

4.1 C语句概述

(四) 空语句

只有一个分号的语句



(什么也不做)

- 用来做流程的转向点
- 用来作为循环语句中的循环体

4.1 C语句概述

(五) 复合语句 用一对 {} 括起来的语句



```
{ z=x+y;  
  t=z/100;  
  printf(“%f”,t);  
}
```

4.2 赋值语句

• 赋值语句是由赋值表达式加上一个分号构成

例： a=100 赋值表达式
 a=100; 赋值语句



4.3 数据输入输出的概念及在C语言中的实现

C语言本身不提供输入输出语句, 输入和输出操作是由C函数库中的函数来实现的

例如:

| | |
|------------------------------|------------------------------|
| 字符输入函数: <code>getchar</code> | 字符输出函数: <code>putchar</code> |
| 格式输入函数: <code>scanf</code> | 格式输出函数: <code>printf</code> |
| 字符串输入函数: <code>gets</code> | 字符串输出函数: <code>puts</code> |



4.3 数据输入输出的概念及在C语言中的实现

在使用系统库函数时, 要用预编译命令

“#include”将有关的“头文件”包括到用户源程序文件中。

例如: 在调用标准输入输出库函数时, 文件开头应该有:

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

或:

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

头文件



4.4 字符数据的输入输出

(一) 字符输出函数

- 一般形式: `putchar (c)`

字符型变量
整型变量

- 函数作用: 向终端输出一个字符



4.4 字符数据的输入输出

例4.1 输出单个字符。

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

运行结果： B
O
Y

```
    putchar(a);putchar('\n');putchar(b);putchar('\n');putchar(c);putchar('\n');
```

运行结果： BOY

4.4 字符数据的输入输出

(二) 字符输入函数

- 一般形式： `getchar ()`
- 函数作用： 从输入设备输入一个字符。
- 函数值： 从输入设备得到的字符。



4.4 字符数据的输入输出

例4.2 输入单个字符。

```
#include<stdio.h>
void main()
{ char c;
  c=getchar();
  putchar(c);
  putchar('\n');
}
```

运行程序：

从键盘输入字符 ‘a’

按Enter键（回车）

屏幕上将显示输出的字符 ‘a’

a ✓

a

4.5 格式输入与输出

程序补充完整：从键盘上输入一个小写字母字符，将它转换为大写字母。

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

```
void main()
```

```
{ char ch;
```

```
ch=getchar();
```

```
_____;
```

```
putchar(_____); }
```

4.5 格式输入与输出

(一) 格式输出函数

- 函数作用：向输出设备输出**若干个任意类型**的数据。
- 一般格式：printf（格式控制，输出表列）

格式控制：是用双撇号括起来的字符串

包括：

- 普通字符
- 格式说明

输出表列：需要输出的数据，可以是表达式

4.5 格式输入与输出

普通字符，包含转义字符

观察以下程序的输出结果

```
#include<stdio.h>
void main()
{printf("The string is: c:\\tools\\'book.txt");
}
```

4.5 格式输入与输出

格式说明:

必须以“%”开头，以9个格式字符之一为结束，中间可以插入附加格式字符（见书81页表4-2的内容）。

%d:以带符号的十进制形式输出整数

%o:以八进制无符号形式输出整数

%x:以十六进制无符号形式输出整数

%u:以无符号十进制形式输出整数

%c:以字符形式输出，只输出一个字符

%s:输出字符串

%f:以小数形式输出单，双精度数，隐含输出六位小数

%e:以指数形式输出实数

%g:选用%f或%e格式中输出宽度较短的一种格式，不输出无意义的0

4.5 格式输入与输出

`%d`:以带符号的十进制形式输出整数

用法:

- ① `%d`:按十进制整型数据的实际长度输出。
- ② `%m d`:`m`为指定的输出字段的宽度。如果数据的位数小于`m`,则左端补以空格,若大于`m`,则按实际位数输出。

以下程序段的输出分别是_____。

```
int x=496;
```

```
printf(" *%-06d*\n",x);
```

- A. *496 * B. * 496* C. *000496* D.输出格式符不合法

```
main()
```

```
{int a=1234;
```

```
printf("%2d\n",a);}
```

- A.12 B.34 C.1234 D.提示出错,无结果



4.5 格式输入与输出

应用:

复合赋值运算

int x=-2,y=2; 则执行表达式 $y+=y-=x*=y$ 后的y值?

可以编程如下

```
#include<stdio.h>
void main()
{
int x=-2,y=2;
y+=y-=x*=y;
printf("%d",y);
}
```



4.5 格式输入与输出

应用:

执行语句“`x=(a=3, b=a--);`”后，`x`、`a`、`b`的值依次为多少？

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

```
void main()
```

```
{
```

```
int a,b,x;
```

```
x=(a=3,b=a--);
```

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

```
}
```



4.5 格式输入与输出

%o: 以八进制无符号形式输出整数

%x: 以十六进制无符号形式输出整数

变量d, o, h为相同值 $(75)_{10}$ ，分别输出该数值的十进制、八进制和十六进制数值

```
#include<stdio.h>
void main()
{int d=o=h=75;
printf("%d, %o, %x\n", d, o, h);
}
```

思考：程序错在什么地方，怎么修改？



4.5 格式输入与输出

整形变量分别赋予不同进制的数值，观察程序的输出结果

提示：第三章中学过，整常数有三种形式表示：十进制，八进制、十六进制

```
#include<stdio.h>
void main()
{int d,o,h;
d=75;
o=075;
h=0x75;
printf("%d, %d, %d\n", d, o, h);
printf("%o, %o, %o\n", d, o, h);
printf("%x, %x, %x\n", d, o, h);}
```



4.5 格式输入与输出

应用

转义字符的使用

- `\ddd ddd`表示1到3位八进制数字
- `\xhh hh`表示1到2位十六进制数字

下面程序错在什么地方？

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

```
void main()
```

```
{printf("%d,%o,%x\n",75,\75,\x75);}
```

转义字符写在格式控制里面

输出语句改写为，`printf("75*\75*\x75");` 输出结果为多少？



4.5 格式输入与输出

`%c`:以字符形式输出，只输出一个字符

一个整数，只要它的值在 $0 \sim 255$ 范围内，可以用“`%c`”使之按字符形式输出，在输出前，系统会将该整数作为ASCII码转换成相应的字符；一个字符数据也可以用整数形式输出。

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

```
void main()
```

```
{printf("%c,%c,%c\n",75,'\75','\x75');
```

```
printf(“%d,%o,%x\n”,‘K’,‘=’,‘u’);
```

```
}
```

输出结果为多少？



4.5 格式输入与输出

%f:以小数形式输出单, 双精度数, 隐含输出六位小数
不指定字段宽度, 由系统自动指定字段宽度, 使整数部分全部输出, 并输出 6 位小数。

应当注意, 在输出的数字中并非全部数字都是有效数字。
例如, 单精度实数的有效位数一般为 7 位。

- `%m.n f`。指定输出的数据共占 m 列, 其中有 n 位小数。
如果数值长度小于 m , 则左端补空格。
- `%-m.n f` 与 `%m.n f` 基本相同,
只是使输出的数值向左端靠, 右端补空格。

以下程序段的输出结果是_____。

```
double a=513.789215;  
printf("a=%8.6f,a=%8.2f,a=%14.8f,a=%14.8lf\n",a,a,a,a);
```



4.5 格式输入与输出

应用:

若已定义x和y为double类型，则表达式“ $x=1,y=x+3/2$ ”的值是多少？注意观察程序的输出结果

```
#include<stdio.h>
void main()
{
double x,y;
printf("%f\n",x=1,y=x+3/2);
printf("%f\n",(x=1,y=x+3/2));
printf("%d\n",(x=1,y=x+3/2));
}
```



4.5 格式输入与输出

%s:输出字符串

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

例4.5字符串的输出。

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

观察程序输出结果

4.5 格式输入与输出

想输出%，则应该在格式控制字符串中用连续两个%表示。

```
#include<stdio.h>
void main()
{
int a=2,b=5;
printf("a=%d,b=%d\n",a,b);
}
```



4.5 格式输入与输出

(二) 格式输入函数

- 函数作用：按照变量在内存的地址将变量值存进去。 **同printf函数**
- 一般格式

是由若干个地址组成的表列，可以是变量的地址，或字符串的首地址

注意：输入数据时不能规定精度。



4.5 格式输入与输出

例4.9 用scanf函数输入数据。

```
#include <stdio.h>
void main ( )
{int a, b, c;
  scanf ("%d%d%d", &a, &b, &c);
  printf ("%d, %d, %d\n", a, b, c);
}
```

a在内存中的地址
&是地址运算符

运行情况：

3 4 5 ↵

(输入a, b, c的值)

3, 4, 5

(输出a, b, c的值)

4.5 格式输入与输出

使用 `scanf` 函数时应注意的问题：

(1) `scanf` 函数中的“格式控制”后面应当是变量地址，而不是变量名。

```
#include<stdio.h>
void main()
{int x;
scanf("%d", x);
printf("%d", x);
}
```

观察程序的输入数据是否为输出数据



4.5 格式输入与输出

使用 `scanf` 函数时应注意的问题：

- (2) 如果在“格式控制”字符串中除了格式说明以外还有其他字符，则在输入数据时在对应位置应输入与这些字符相同的字符。

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



4.5 格式输入与输出

使用 `scanf` 函数时应注意的问题：

(3) 在用“%c”格式输入字符时，空格字符和“转义字符”都作为有效字符输入

```
#include<stdio.h>
void main()
{char x,y;
scanf("%c%c",&x,&y);
printf("%d %d",x,y);
}
```

输入数据1：a空格回车

输入数据2：a回车

查看输出结果



4.5 格式输入与输出

使用 `scanf` 函数时应注意的问题：

(4) 在输入数据时，遇以下情况时认为该数据结束。

① 遇空格，或按“回车”或“跳格”（Tab）键；

```
#include <stdio.h>
void main ( )
{int a,b,c;
 scanf ("%d%d", &a, &b);
 printf ("%d, %d\n", a, b);
}
```



4.5 格式输入与输出

使用 `scanf` 函数时应注意的问题：

(4) 在输入数据时，遇以下情况时认为该数据结束。

② 按指定的宽度结束，如“%3d”，只取3列；

```
#include <stdio.h>
void main ( )
{int a,b,c;
scanf("%3d",&a);
printf("%d\n",a);
}
```

输入数据为12345，观察输出结果



4.5 格式输入与输出

使用 `scanf` 函数时应注意的问题：

(4) 在输入数据时，遇以下情况时认为该数据结束。

③ 遇非法输入。

```
#include <stdio.h>
void main ( )
{int a,b,c;
scanf("%d",&a);
printf("%d\n",a);
}
```

输入123abc，观察输出结果



4.5 格式输入与输出

如果在%后有一个“*”附加说明符，表示跳过它指定的列数。

```
#include<stdio.h>
void main()
{int a,b,c;
scanf("%d%*d%d%d",&a,&b,&c);
printf("%d %d %d",a,b,c);}
```

当输入以下数据时，100 200 300 400 500<回车>
观察输出结果



4.6 顺序结构程序设计举例

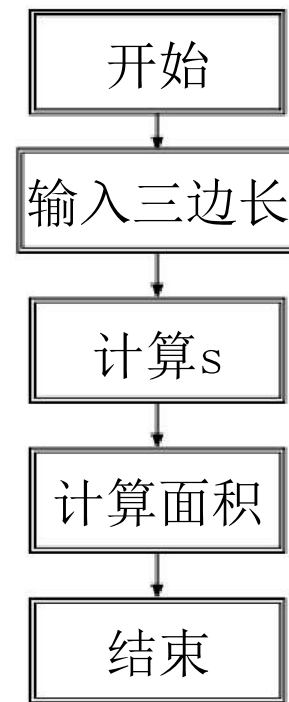
例4.10 输入三角形的三边长，求三角形面积。

假设：三个边长 a ， b ， c 能构成三角形。

已知面积公式：

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

$$s = (a+b+c) * 0.5$$



4.6 顺序结构程序设计举例

```
#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));
    printf("a=%7.2f, b=%7.2f, c =%7.2f, s
    =%7.2f\n", a, b, c, s);
    printf("area=%7.2f\n", area);
}
```