

第4章 顺序结构



§ 1 赋值语句

赋值表达式

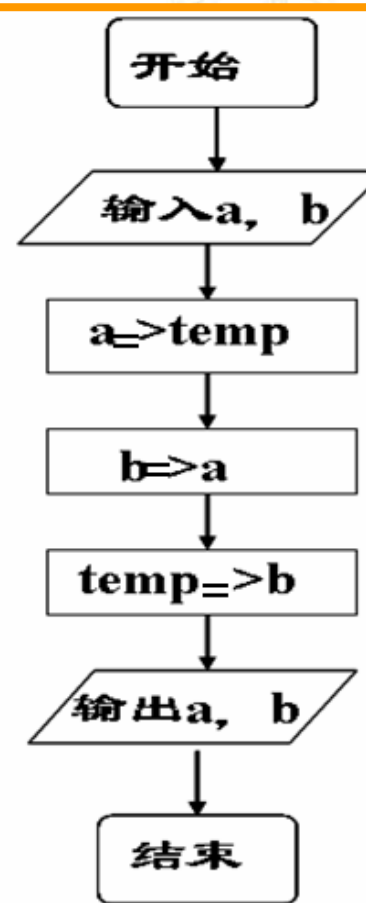
变量=表达式

赋值语句

变量=表达式;

例:编程交换 a, b 二整型变量之值

```
main()  
{ int a , b , temp ;  
  scanf("%d,%d",&a,&b);  
  printf("Before swapped , a=%d,b=%d\n",a,b);  
  temp=a;  
  a=b;  
  b=temp;  
  printf("After swapped , a=%d,b=%d\n",a,b);  
}
```



§ 2 库函数

一、库函数简介

分类 {

- 数学库函数：实现常用的数学函数计算
- 字符库函数：实现对字符数据的常用处理
- 字符串库函数：实现对字符串数据的常用处理
- I/O库函数：实现I/O操作及对数据文件的处理

eg1: 求 $\sin 19^\circ$ \longrightarrow $\sin(19*3.14/180)$

eg2: 求 $\sin 2$ \longrightarrow $\sin((\text{double})2)$ 或 $\sin(2.0)$

eg3: $\frac{e^x + e^{-x}}{2}$ \longrightarrow $(\exp(x) + \exp(-x))/2.0$



§ 2 库函数



一、库函数简介

分类 {

- 数学库函数：实现常用的数学函数计算
- 字符库函数：实现对字符数据的常用处理
- 字符串库函数：实现对字符数据的常用处理
- I/O库函数：实现I/O操作及对数据文件的处理

eg4: $\frac{2 \ln x}{\lg y + 3b} \longrightarrow 2*\log(x)/(\log10(y)+3*b)$

eg5: $x^7 + \sqrt{x^2 + y^2} \longrightarrow \text{pow}(x, 7.0) + \text{sqrt}(x*x + y*y)$

eg6: $|x^3 + \ln x| \longrightarrow \text{fabs}(\text{pow}(x, 3.0) + \log(x))$



§ 2 库函数

一、库函数简介

分类 {

- 数学库函数：实现常用的数学函数计算
- 字符库函数：实现对字符数据的常用处理
- 字符串库函数：实现对字符串数据的常用处理
- I/O库函数：实现I/O操作及对数据文件的处理

eg7(考题):数学表达式 $\sin 25^\circ + x \cos 30^\circ$ 的C语言表达式_____。

$\sin(25*3.14/180)+x*\cos(30*3.14/180)$

eg8(考题): 设有说明: `double y=0.5,z=1.5;int x=10;`则能够正确使用C语言库函数的赋值语句是_____。

A. `z=exp(y)+fabs(x);` B. `y=log10(y)+pow(y);`
C. `z=sqrt(y-z);` D. `x=(int)(pow((double)x,y)+exp(y-0.2));`

D

§ 2 库函数

一、库函数简介

分类 {

- 数学库函数：实现常用的数学函数计算
- 字符库函数：实现对字符数据的常用处理
- 字符串库函数：实现对字符串数据的常用处理
- I/O库函数：实现I/O操作及对数据文件的处理

eg9：若c中是小写字母，则将其转变为对应的大写字母

```
if(c>='a' && c<='z') c=c-('a'-'A');
```

```
if(islower(c)) c=toupper(c);
```

eg10：将数字字符c转化为对应数字

```
if(c>='0' && c<='9') c=c-'0';
```

```
if(isdigit(c)) c=c-'0';
```



§ 2 库函数

二、文件包含命令

命令格式

`#include <文件名>` 或 `#include "文件名"`

功能

将所包含的文件内容在`#include`命令处宏展开

分类

数学库函数`math.h`: `#include <math.h>`

字符库函数`ctype.h`: `#include <ctype.h>`

字符串库函数`string.h`: `#include <string.h>`

I/O库函数`stdio.h`: `#include <stdio.h>`

例:计算 $\sin 2$ 之值

```
#include<math.h>
#include <stdio.h>
main( )
{ printf("%f\n",sin(2.0));
}
```




§ 3 字符I/O函数

—文件包含：#include <stdio.h>

一、字符输出函数

● 函数说明

```
int putchar(char c);
```

● 函数调用格式

```
putchar(c)
```

- **功能:** 输出c所表示的字符。若输出正确，返回值为输出的字符；若出错，返回EOF (#define EOF -1)

```
eg1: putchar('a');
```

```
eg3: putchar(97);
```

```
eg6: putchar('\141');
```

```
eg2: char c='a';  
      putchar(c);
```

```
eg4: putchar(0141);
```

```
eg7: putchar('\x61');
```

```
eg5: putchar(0x61);
```




§ 3 字符I/O函数

—文件包含：#include <stdio.h>

例：分析程序的运行结果。

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

```
main( )
```

```
{ char a='B',b='O',c='Y';
```

```
    putchar('B'); putchar('O'); putchar('Y'); putchar('\n');
```

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

```
    putchar(66); putchar(79); putchar(89);  putchar(10);
```

```
    putchar('\102'); putchar('\117');
```

```
    putchar('\131'); putchar('\12');
```

```
    putchar('\x42'); putchar('\x4f');
```

```
    putchar('\x59'); putchar('\n');
```

```
}
```

BOY
BOY
BOY
BOY
BOY



§ 3 字符I/O函数

—文件包含：#include <stdio.h>

二、字符输入函数

● 函数说明

```
int getchar( );
```

● 函数调用格式

```
getchar( )
```

● **功能：**获得标准终端输入流中的当前字符。返回值为所读字符；若出错，返回EOF

例1:分析下列程序的运行结果

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

```
main( )
```

```
{ char c1,c2,c3;
```

```
  c1=getchar( );  c2=getchar( );  c3=getchar( );
```

```
  printf("c1=%c,c2=%c,c3=%c\n",c1,c2,c3); }
```



§ 3 字符I/O函数

—文件包含：#include <stdio.h>

二、字符输入函数

● 函数说明

```
int getchar( );
```

● 函数调用格式

```
getchar( )
```

● **功能：**获得标准终端输入流中的当前字符。返回值为所读字符；若出错，返回EOF

例2:若输入为6，试分析下列程序的运行结果

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

6,54

♠,6

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



§ 4 格式I/O函数

一、格式输出函数

1. 函数调用语句

`printf(格式控制串[,参数1, 参数2, ...参数n]);`

2. 功能: 计算各参数之值, 按格式控制串指定的格式加以输出

3. 格式控制串: 由双引号定界, 内含ASCII码字符和格式说明
——ASCII码字符, 原样输出

——格式说明, 控制后继一个参数值的输出格式

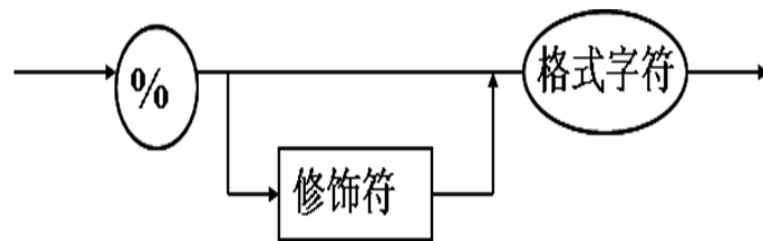
4. 格式说明语法

eg1: `printf("program.\n");`

eg2: `printf("%d,%d",3,b);`

eg3: `printf("%d+%d",a+1,b+2);`

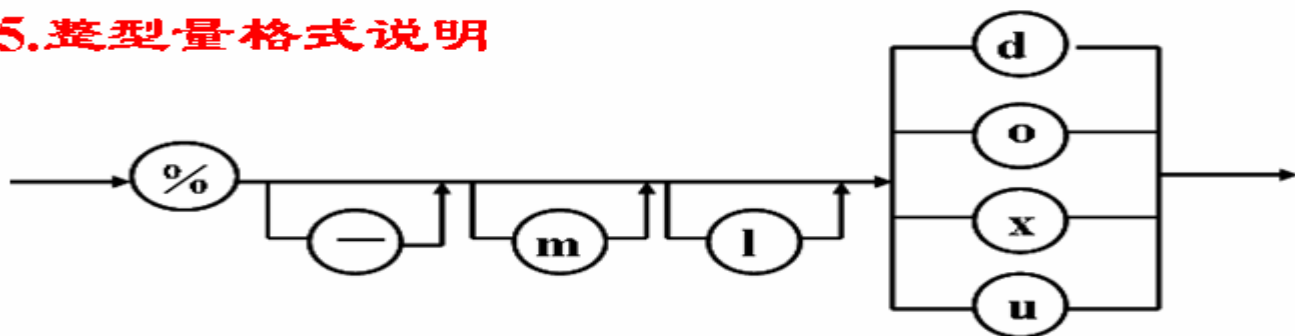
eg4: `printf("a=%d,b=%d\n",a+1,b+2);`



§ 4 格式I/O函数

一、格式输出函数

5. 整型量格式说明



d: 将数据以十进制整数形式输出

o: 将数据看成无符号整数，将其值以八进制形式输出

x: 将数据以无符号十六进制整数形式输出

u: 将数据以无符号十进制整数形式输出

eg1: `int a=5,b=-8;`

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

a=5,b=-8

eg2: `int a=17,b=-1;`

`printf("a=0%o, b=0%o\n",a,b);`

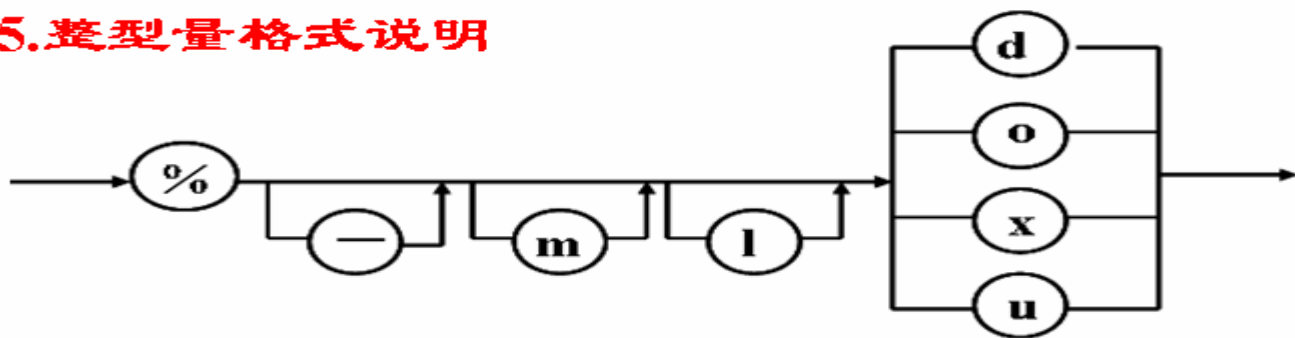
a=021,b=0177777



§ 4 格式I/O函数

一、格式输出函数

5. 整型量格式说明



d: 将数据以十进制整数形式输出

o: 将数据看成无符号整数，将其值以八进制形式输出

x: 将数据以无符号十六进制整数形式输出

u: 将数据以无符号十进制整数形式输出

eg3:
main()
{ int b=-1;
printf(“b:\ndec=%**d**\noct=%**o**\nhex=%**x**\n”,b,b,b); }

b:

dec=-1

oct=177777

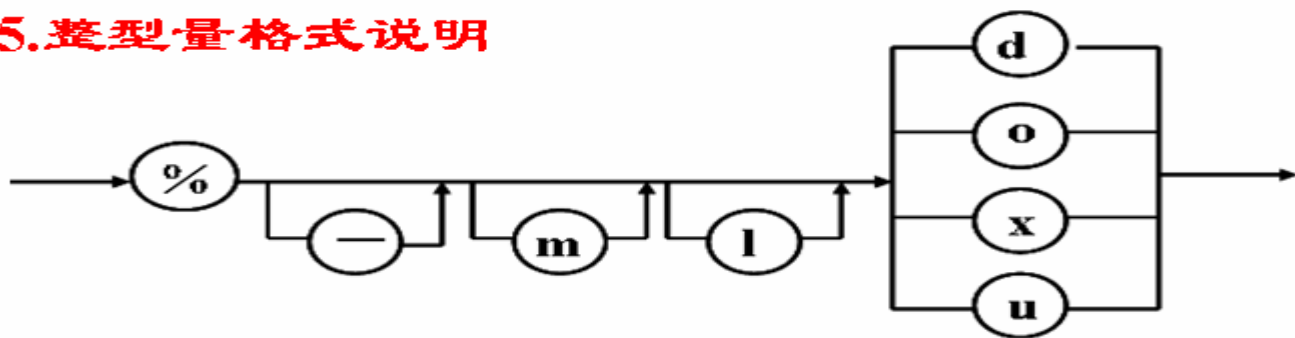
hex=ffff



§ 4 格式I/O函数

一、格式输出函数

5. 整型量格式说明



d: 将数据以十进制整数形式输出

o: 将数据看成无符号整数，将其值以八进制形式输出

x: 将数据以无符号十六进制整数形式输出

u: 将数据以无符号十进制整数形式输出

eg4: unsigned a=65535;

int b=-2;

printf("a=%**u**,b=%**u**\n",a,b);

printf("a=%**d**,b=%**d**\n",a,b);

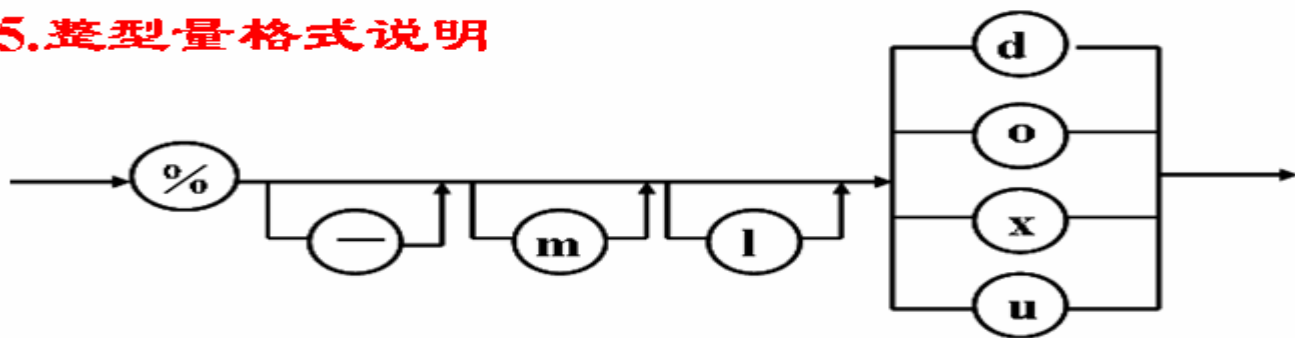
a=65535,b=65534

a=-1,b=-2

§ 4 格式I/O函数

一、格式输出函数

5. 整型量格式说明



- **修饰符m:** 数据的输出域宽
- **修饰符-:** 在输出域内左对齐
- **修饰符l:** 输出长整型数据

eg5: `a=3;`
`printf("%d,%3d\n",a,a);`

3, 3

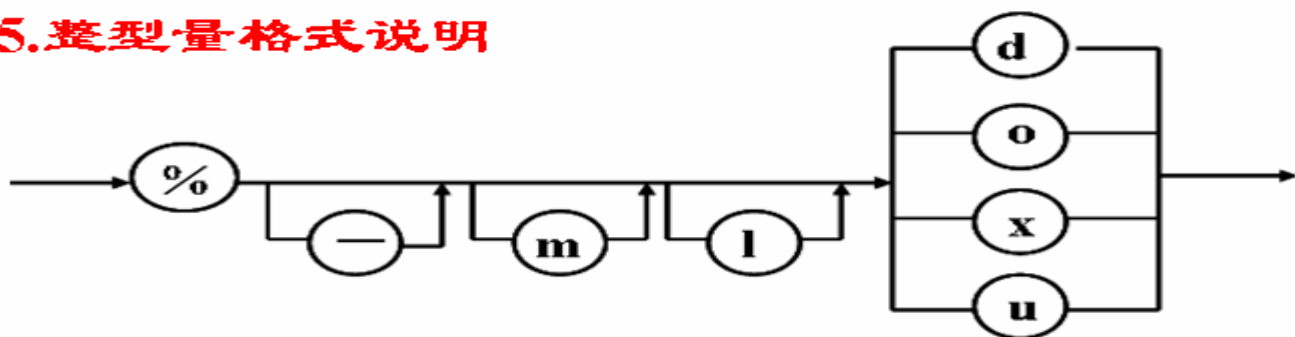
eg6: `a=3;`
`printf("%d,%3d,%-3d\n",a,a,a);`

3, 3, 3

§ 4 格式I/O函数

一、格式输出函数

5. 整型量格式说明



- **修饰符m:** 数据的输出域宽
- **修饰符-:** 在输出域内左对齐
- **修饰符l:** 输出长整型数据

```
eg7: long a;  
      a=135790;  
      printf("a=%d\n",a);
```

错误

```
eg7: long a;  
      a=135790;  
      printf("a=%ld\n",a);
```

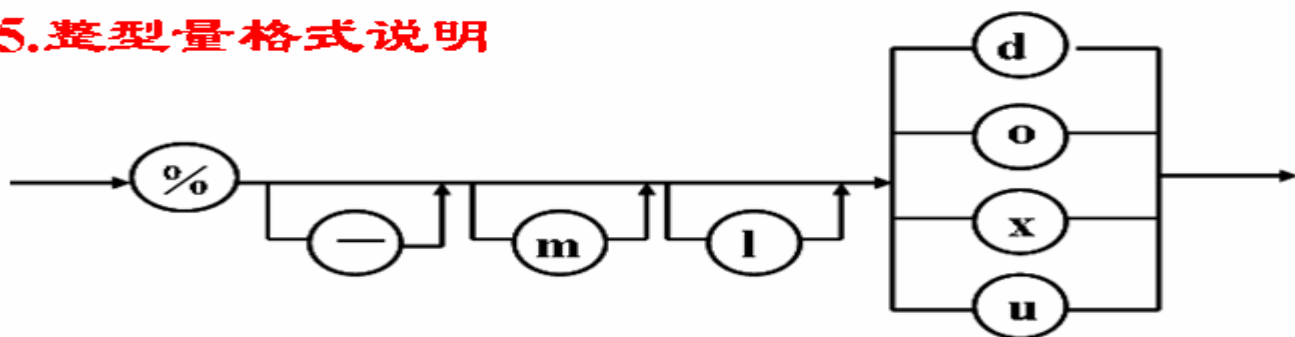
正确

a=135790

§ 4 格式I/O函数

一、格式输出函数

5. 整型量格式说明



- **修饰符m:** 数据的输出域宽
- **修饰符-:** 在输出域内左对齐
- **修饰符l:** 输出长整型数据

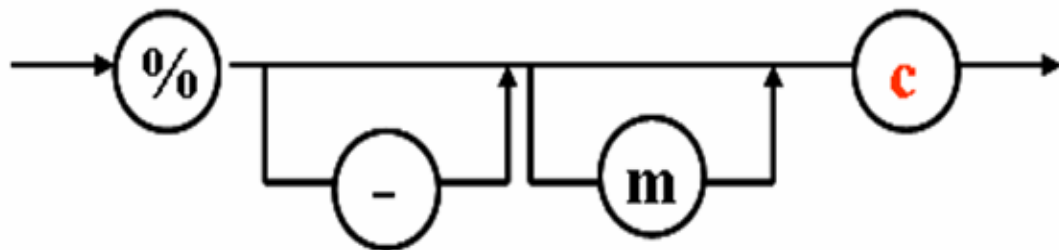
```
eg8:  main(  
      { long a=135790;  
        printf("%ld,%8ld,%-9ld\n",a,a,a); }  
      }
```

135790, 135790, 135790

§ 4 格式I/O函数

一、格式输出函数

6. 字符数据的格式说明



eg: main()

```
{ char c='x';
```

```
    printf("%c,%3c,%-3c\n",'x','x','x');
```

```
    printf("%c,%3c,%-3c\n",c,c,c);
```

```
}
```

X, X, X

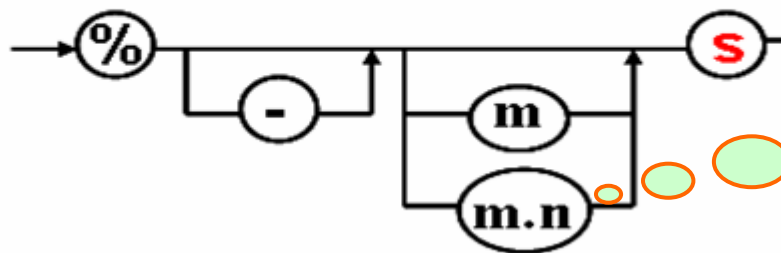
X, X,X



§ 4 格式I/O函数

一、格式输出函数

7. 字符串数据的格式说明



截取串左侧
n个字符

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

```
main( )
```

```
{
```

```
    printf("%s,%07s,%-7s\n","china","china","china");
```

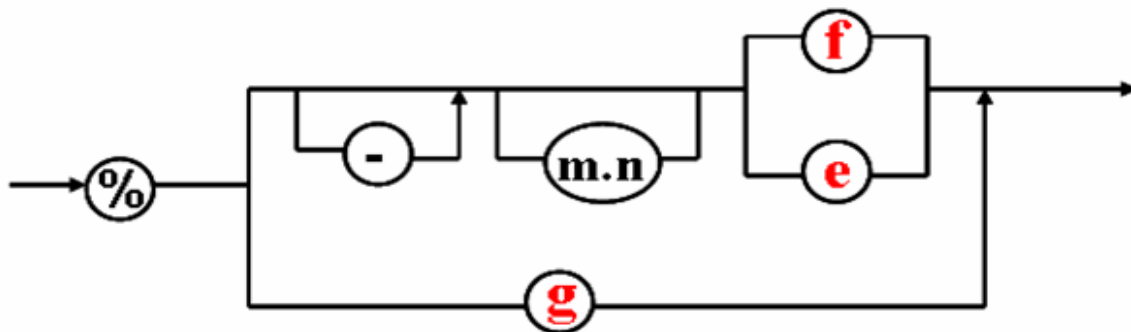
```
    printf("%05.2s,%0.3s,%-5.2s\n","china","china","china");
```

```
}
```

§ 4 格式I/O函数

一、格式输出函数

8. 实型数据的格式说明



.n为精度域宽，即小数部分位数,系统默认值为6

f:将单精度或双精度实数以十进制小数形式输出

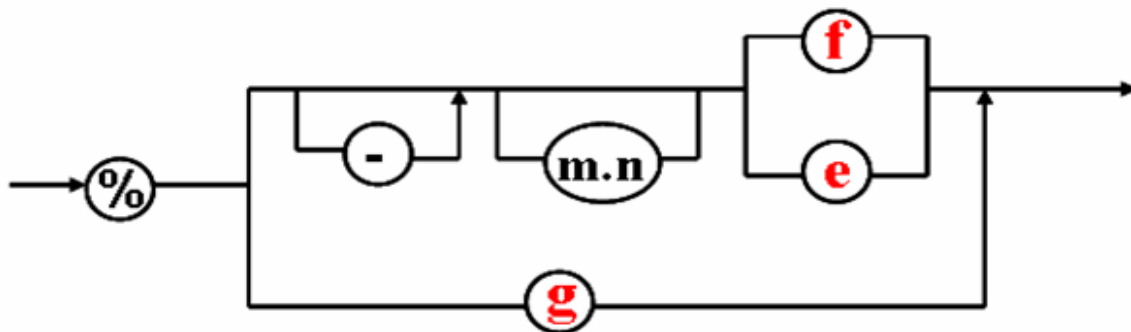
```
eg1: float a=33.79;  
      printf("%f,%7.2f,%.2f",a,a,a);
```

33.789978, 33.79, 33.79

§ 4 格式I/O函数

一、格式输出函数

8. 实型数据的格式说明



.n为精度域宽，即小数部分位数,系统默认值为6

f:将单精度或双精度实数以十进制小数形式输出

eg2: **double** x,y;

x=11111111111111.11111111; /***22位***/

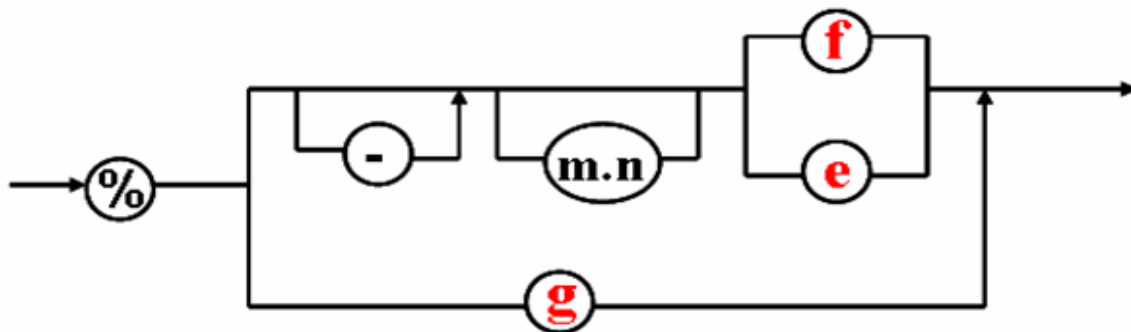
y=22222222222222.22222222; printf("%**f**",x+y);

33333333333333.333**010**

§ 4 格式I/O函数

一、格式输出函数

8. 实型数据的格式说明



e:将单精度或双精度实数以**规范化指数**形式输出
 $[\pm]n.nnnnnne \pm mmm$ —— $1 \leq \text{尾数} \leq 10$

eg3:main()

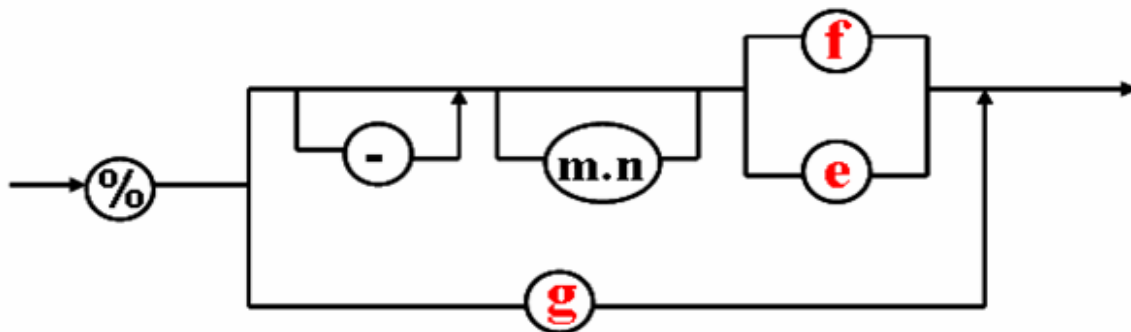
```
{ printf("%e,%10.2e\n",123.456,123.456 ); }
```

1.234560e+002, 1.23e+002

§ 4 格式I/O函数

一、格式输出函数

8. 实型数据的格式说明



eg4:main()

```
{ float f=123.468;  
  printf("%f,%e,%g",f,f,f); }
```

123.468000 , 1.234680e+002 , 123.468



§ 4 格式I/O函数

例1:试分析下列程序的运行结果

```
#include <stdio.h>

main( )
{ int a=-10,b=-3;
  printf("%d\t",a%b);
  printf("%d\t",a/b*b);
  printf("%d\t",-a%b);
  printf("%d\n",a-=b+++1);
}
```

-1	-9	1	-8
----	----	---	----



§ 4 格式I/O函数

例2:输入一个字符，试求出其前驱字符和后继字符，并按ASCII码值由小至大次序输出这三个字符及其ASCII码值。

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

```
main( )
```

```
{ char c,c1,c2;
```

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

```
  c1=c-1,c2=c+1;
```

```
  printf("Character:%c %c %c\n",c1,c,c2);
```

```
  printf("ASCII:%d %d %d\n",c1,c,c2);
```

```
}
```



§ 4 格式I/O函数

二、格式输入函数

1. 函数调用语句

`scanf(格式控制串, 变量1地址[, 变量2地址, ...变量n地址]);`

——&:取地址运算符

2. 功能:从标准输入上读入字符序列, 按格式控制进行解释, 再将结果送入变量的内存单元

——各数据以空白格字符(空格、制表、回车)分隔

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

输入: 1 2<回车>

eg: `scanf("%c%c",&c1,&c2);`

输入: 1 <Tab>2<回车>

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

输入: 1 <Enter>
2<Enter>



§ 4 格式I/O函数

二、格式输入函数

1. 函数调用语句

`scanf(格式控制串, 变量1地址[, 变量2地址, ... 变量n地址]);`

——&:取地址运算符

2. 功能:从标准输入上读入字符序列, 按格式控制进行解释, 再将结果送入变量的内存单元

——各数据以空白格字符(空格、制表、回车)分隔

3. 格式字符

格式字符	功能
d	用以输入十进整数
o	用以输入8进整数
x	用以输入16进整数
f(e)	用以输入单精度实数
c	用以输入单个字符
s	用以输入字符串

希望a内存值为12

eg1: `scanf("%d",&a);`

输入: 12<Enter>



§ 4 格式I/O函数

二、格式输入函数

1. 函数调用语句

`scanf(格式控制串, 变量1地址[, 变量2地址, ...变量n地址]);`

——&:取地址运算符

2. 功能:从标准输入上读入字符序列, 按格式控制进行解释, 再将结果送入变量的内存单元

——各数据以空白格字符(空格、制表、回车)分隔

3. 格式字符

格式字符	功能
d	用以输入十进整数
o	用以输入8进整数
x	用以输入16进整数
f(e)	用以输入单精度实数
c	用以输入单个字符
s	用以输入字符串

希望a内存值为11

eg2: `scanf("%o",&a);`

输入: 013<Enter>

13<Enter>



§ 4 格式I/O函数

二、格式输入函数

1. 函数调用语句

`scanf(格式控制串, 变量1地址[, 变量2地址, ...变量n地址]);`

——&:取地址运算符

2. 功能:从标准输入上读入字符序列, 按格式控制进行解释, 再将结果送入变量的内存单元

——各数据以空白格字符(空格、制表、回车)分隔

3. 格式字符

格式字符	功能
d	用以输入十进整数
o	用以输入8进整数
x	用以输入16进整数
f(e)	用以输入单精度实数
c	用以输入单个字符
s	用以输入字符串

希望a内存值为-11

eg3: `scanf("%x",&a);`

输入: `-0xb<Enter>`

`-b<Enter>`



§ 4 格式I/O函数

二、格式输入函数

1. 函数调用语句

`scanf(格式控制串, 变量1地址[, 变量2地址, ... 变量n地址]);`

——&:取地址运算符

2. 功能:从标准输入上读入字符序列, 按格式控制进行解释, 再将结果送入变量的内存单元

——各数据以空白格字符(空格、制表、回车)分隔

3. 格式字符

格式字符	功能
d	用以输入十进整数
o	用以输入8进整数
x	用以输入16进整数
f(e)	用以输入单精度实数
c	用以输入单个字符
s	用以输入字符串

希望x为12.35

`eg4:scanf("%f",&x);`

等价于

`eg4:scanf("%e",&x);`



§ 4 格式I/O函数

二、格式输入函数

1. 函数调用语句

`scanf(格式控制串,变量1地址[, 变量2地址, ...变量n地址]);`

——&:取地址运算符

2. 功能:从标准输入上读入字符序列, 按格式控制进行解释, 再将结果送入变量的内存单元

——各数据以空白格字符(空格、制表、回车)分隔

3. 格式字符

格式字符	功能
d	用以输入十进整数
o	用以输入8进整数
x	用以输入16进整数
f(e)	用以输入单精度实数
c	用以输入单个字符
s	用以输入字符串

```
eg5:char c;  
scanf("%c",&c);
```

输入: 0123<回车>

c的内存值为'0'

§ 4 格式I/O函数

二、格式输入函数

1. 函数调用语句

`scanf(格式控制串,变量1地址[, 变量2地址, ...变量n地址]);`

——&:取地址运算符

2. 功能:从标准输入上读入字符序列, 按格式控制进行解释, 再将结果送入变量的内存单元

——各数据以空白格字符(空格、制表、回车)分隔

3. 格式字符

格式字符	功能
d	用以输入十进整数
o	用以输入8进整数
x	用以输入16进整数
f(e)	用以输入单精度实数
c	用以输入单个字符
s	用以输入字符串

```
eg6:char str[6];  
scanf("%s",str);
```

输入: China<回车>

str

str[6]

C	h	i	n	a	\0
---	---	---	---	---	----



§ 4 格式I/O函数

二、格式输入函数

1. 函数调用语句

`scanf(格式控制串, 变量1地址[, 变量2地址, ...变量n地址]);`

——&:取地址运算符

2. 功能:从标准输入上读入字符序列, 按格式控制进行解释, 再将结果送入变量的内存单元

——各数据以空白格字符(空格、制表、回车)分隔

3. 格式字符

4. 修饰符

修饰符	说明
m	指示所接受的数据域宽度
l	用以输入长整型 (%ld,%lo,%lx)或double 型(%lf,%le)

eg1: `scanf("%3d%d",&a,&b);`

输入: 12345<CR>

内存值: a=123 b=45



§ 4 格式I/O函数

二、格式输入函数

1. 函数调用语句

`scanf(格式控制串, 变量1地址[, 变量2地址, ...变量n地址]);`

——&:取地址运算符

2. 功能:从标准输入上读入字符序列, 按格式控制进行解释, 再将结果送入变量的内存单元

——各数据以空白格字符(空格、制表、回车)分隔

3. 格式字符

4. 修饰符

修饰符	说明
m	指示所接受的数据域宽度
l	用以输入长整型 (%ld,%lo,%lx)或double 型(%lf,%le)

eg1: `scanf("%3d%d",&a,&b);`

输入: 1 45<CR>

内存值: a=1 b=45



§ 4 格式I/O函数

二、格式输入函数

1. 函数调用语句

`scanf(格式控制串, 变量1地址[, 变量2地址, ...变量n地址]);`

——&:取地址运算符

2. 功能:从标准输入上读入字符序列, 按格式控制进行解释, 再将结果送入变量的内存单元

——各数据以空白格字符(空格、制表、回车)分隔

3. 格式字符

4. 修饰符

修饰符	说明
m	指示所接受的数据域宽度
l	用以输入长整型 (%ld,%lo,%lx)或double 型(%lf,%le)

eg1: `scanf("%3d%d",&a,&b);`

输入: 12<tab>345<CR>

内存值: a=12 b=345



§ 4 格式I/O函数

二、格式输入函数

1. 函数调用语句

`scanf(格式控制串, 变量1地址[, 变量2地址, ... 变量n地址]);`

——&:取地址运算符

2. 功能:从标准输入上读入字符序列, 按格式控制进行解释, 再将结果送入变量的内存单元

——各数据以空白格字符(空格、制表、回车)分隔

3. 格式字符

4. 修饰符

修饰符	说明
m	指示所接受的数据域宽度
l	用以输入长整型 (%ld,%lo,%lx)或double 型(%lf,%le)

eg1: `scanf("%3d%d",&a,&b);`

输入: 12 <CR>
34 <CR>

内存值: a=12 b=34



§ 4 格式I/O函数

二、格式输入函数

1. 函数调用语句

`scanf(格式控制串, 变量1地址[, 变量2地址, ...变量n地址]);`

——&:取地址运算符

2. 功能:从标准输入上读入字符序列, 按格式控制进行解释, 再将结果送入变量的内存单元

——各数据以空白格字符(空格、制表、回车)分隔

3. 格式字符

4. 修饰符

修饰符	说明
m	指示所接受的数据域宽度
l	用以输入长整型 (%ld,%lo,%lx)或double 型(%lf,%le)

eg2: `char c1,c2;`
`scanf("%3c%c",&c1,&c2);`

输入: 1234<CR>

内存值: c1='1' c2='4'



§ 4 格式I/O函数

二、格式输入函数

1. 函数调用语句

`scanf(格式控制串,变量1地址[, 变量2地址, ...变量n地址]);`

——&:取地址运算符

2. 功能:从标准输入上读入字符序列, 按格式控制进行解释, 再将结果送入变量的内存单元

——各数据以空白格字符(空格、制表、回车)分隔

3. 格式字符

4. 修饰符

修饰符	说明
m	指示所接受的数据域宽度
l	用以输入长整型(%ld,%lo,%lx)或double型(%lf,%le)

```
eg3: char str[6];  
      scanf("%5c",str);
```

```
eg3: char str[6];  
      scanf("%s",str);
```

不等价



§ 4 格式I/O函数

二、格式输入函数

1. 函数调用语句

`scanf(格式控制串, 变量1地址[, 变量2地址, ... 变量n地址]);`

——&: 取地址运算符

2. 功能: 从标准输入上读入字符序列, 按格式控制进行解释, 再将结果送入变量的内存单元

——各数据以空白格字符(空格、制表、回车)分隔

3. 格式字符

4. 修饰符

修饰符	说明
m	指示所接受的数据域宽度
l	用以输入长整型(%ld,%lo,%lx)或double型(%lf,%le)

eg4: `int i; char c;`
`scanf("%4c%d",&c,&i);`

输入: a1112<CR>

内存值: c='a' i=2



§ 4 格式I/O函数

二、格式输入函数

1. 函数调用语句

`scanf(格式控制串, 变量1地址[, 变量2地址, ... 变量n地址]);`

——&:取地址运算符

2. 功能:从标准输入上读入字符序列, 按格式控制进行解释, 再将结果送入变量的内存单元

——各数据以空白格字符(空格、制表、回车)分隔

3. 格式字符

4. 修饰符

修饰符	说明
m	指示所接受的数据域宽度
l	用以输入长整型(%ld,%lo,%lx)或double型(%lf,%le)

```
eg5: long l;  
      scanf("%d",&l);
```

错误

```
eg5: long l;  
      scanf("%ld",&l);
```

正确



§ 4 格式I/O函数

二、格式输入函数

1. 函数调用语句

`scanf(格式控制串, 变量1地址[, 变量2地址, ...变量n地址]);`

——&:取地址运算符

2. 功能:从标准输入上读入字符序列, 按格式控制进行解释, 再将结果送入变量的内存单元

——各数据以空白格字符(空格、制表、回车)分隔

3. 格式字符

4. 修饰符

修饰符	说明
m	指示所接受的数据域宽度
l	用以输入长整型 (%ld,%lo,%lx)或double 型(%lf,%le)

eg6: `double f1;
scanf("%f",&f1);`

错误

eg6: `double f1;
scanf("%lf",&f1);`

正确



§ 4 格式I/O函数

二、格式输入函数

5.如何正确使用数据分隔符？



用空白格字符分隔各个数据,但不能分隔字符

数据



指定宽度自动分隔各个数据

分隔符



在scanf语句中给出特定分隔符



以非法输入结束上一数据

```
eg1: int i,j,k;  
scanf("%d%d%d",&i,&j,&k);
```

```
eg3: char s1[5],s2[5];  
scanf("%s%s",s1,s2);
```

```
eg2: float x,y;  
scanf("%f%f",&x,&y);
```

```
eg4: char c1,c2,c3;  
scanf("%c%c%c",&c1,&c2,&c3);
```


§ 4 格式I/O函数

二、格式输入函数

5.如何正确使用数据分隔符？



用空白格字符分隔各个数据,但不能分隔字符

数据



指定宽度自动分隔各个数据

分隔符



在scanf语句中给出特定分隔符



以非法输入结束上一数据

```
eg5: int a,b,c;  
scanf("%2d%3d%d",&a,&b,&c);
```

```
eg6: char c1,c2,c3;  
scanf("%3c%3c%c",&c1,&c2,&c3);
```



§ 4 格式I/O函数

二、格式输入函数

5.如何正确使用数据分隔符？



用空白格字符分隔各个数据,但不能分隔字符

数据



指定宽度自动分隔各个数据

分隔符



在scanf语句中给出特定分隔符



以非法输入结束上一数据

```
eg7: int a,b,c;  
scanf("%d%d%d",&a,&b,&c);
```

```
eg8: int a,b,c;  
scanf("%d,%d,%d",&i,&j,&k);
```

```
eg9: int a,b,c;  
scanf("%d:%d:%d",&a,&b,&c);
```

§ 4 格式I/O函数

二、格式输入函数

5.如何正确使用数据分隔符？



用空白格字符分隔各个数据,但不能分隔字符

数据



指定宽度自动分隔各个数据

分隔符



在scanf语句中给出特定分隔符



以非法输入结束上一数据

```
eg10: int a,b,c;  
scanf("a=%d,b=%d,c=%d",&a,&b,&c);
```

```
eg11: printf("Input a,b,c:"); /*给出屏幕提示*/  
scanf("%d%d%d",&a,&b,&c);
```

```
eg12: int a,b,c;  
scanf("%d%d%d\n",&a,&b,&c);
```

§ 4 格式I/O函数

二、格式输入函数

5.如何正确使用数据分隔符？



用空白格字符分隔各个数据,但不能分隔字符

数据



指定宽度自动分隔各个数据

分隔符



在scanf语句中给出特定分隔符



以非法输入结束上一数据

```
eg13: scanf("%d%c%f",&i,&c,&x);
```

输入: 1234a123q.26<回车>

内存值: i=1234 c='a' x=123.0



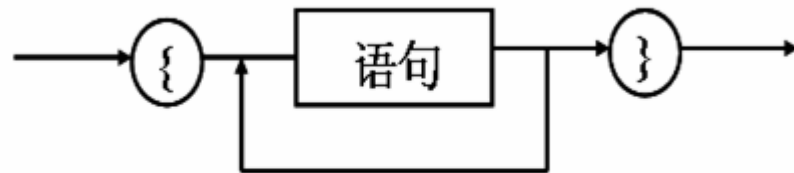
§ 5 C语句的常见类型

● 函数调用语句

eg1: `getchar();`

eg2: `printf("%d",a);`

eg3: `fun();`



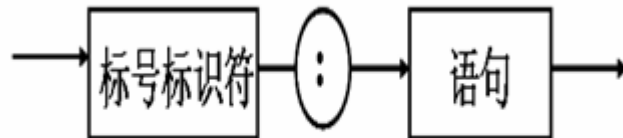
复合语句语法图

● 空语句 ;

——常用作构造**空循环体**等程序设计技

● 复合语句

——复合语句和简单语句具有同等语法地



带标号语句语法图

● 带标号的语句

——标识语句位置，以便于控制转





§ 5 C语句的常见类型

例1:假设 Δabc 的边长能构成三角形, 编程求其面积

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

main( )
{ float a,b,c,s,area;
  scanf("%f,%f,%f",&a,&b,&c);
  s=(a+b+c)/2.0;
  area=sqrt(s*(s-a)*(s-b)*(s-c));
  printf("\na=%.2f,b=%.2f,c=%.2f",a,b,c);
  printf("\narea=%.2f\n",area);
}
```




§ 5 C语句的常见类型

例2:有三门课程成绩(整型), 编程求平均成绩并将其四舍五入

```
#include <stdio.h>

main( )
{ int score1,score2,score3,sum,ave;
  printf("\n Input score,Please!\n");
  scanf("%d,%d,%d",&score1,&score2,&score3);
  sum=score1+score2+score3;
  ave=sum/3.0+0.5;
  printf("\nscores:%d,%d,%d\n",score1,score2,score3);
  printf("Average=%d\n",ave);
}
```