C语言与程序设计

The C Programming Language

第3章 格式化输入与输出

武汉光电国家研究中心信息存储研究部

李春花

主要内容

- C语言提供了一些标准的输入/出函数——系统函数

  putchar、puts、printf

  getchar、gets、scanf

注意：用这些函数时,须用预编译指令:

#include <stdio.h>

3.1 字符输出与输入

- **putchar** ：字符输出函数
- **getchar**：字符输入函数

字符输出函数putchar

函数原型： int putchar(int);

函数功能： 向标准输出设备(显示器)输出一个字符，函数正确执行时返回该字符码，否则返回EOF。

调用方式： putchar (c) ;

↑ char 或 int

举例： 欲输出字符 A：

```
putchar('A'); //常用方式
putchar( 65 );
putchar ('\x41');
```

字符输入函数getchar

函数原型：int getchar(void) ;

功能：从标准输入设备(键盘)读取一个字符，并将该字符作为函数的值。

调用方式: getchar( );

举例：

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

嵌套调用

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

或

```
putchar(getchar( ));
```

函数getchar用法分析

【例3.1】

```
int main()
{
    char c1,c2;
    printf("please input a char:\n");
    c1=getchar();  // c1='a'
    //getchar();

    printf("please input an another char:\n");
    c2=getchar();  // 从输入缓冲区取回车符赋给c2
                   // 即c2='\n'
    printf("%c,%c\n",c1,c2);
    printf("%d,%d\n",c1,c2);

    return 0;
}
```

```
please input a char:
a↙
please input an another char:
a,

97,10
```

getchar( )可用于清空缓冲区中的回车符 '\n'

【例3.4】

```
while( getchar() != EOF )
{
 //...
}
```

行首输入Ctrl+Z↙，用getchar( )判断是否结束输入

3.2 格式化输入与输出

- **printf**：*格式化输出*函数
- **scanf**：*格式化输入*函数

函数printf

printf是C语言中使用得最多的一种输出函数，

它可一次按格式输出多个不同类型的数据。

printf (“•••”，参数1，参数2，•••);

格式控制串

1. 普通字符：原样输出

2. 格式说明：由%开始，转换字符结尾，如：%f, %d, %hu 等

```
printf (“This is the first program.\n”);
int x=10;  float y=10.512;
printf (“x=%d, y=%.1f ”, x, y);
```

```
This is the first program.
x=10,  y=10.5
```

格式说明的基本组成

printf (“ • • • ” ， 参数1，参数2，• • •) ;//由%开始，转换字符结尾

% [特征符] [域宽] [长度修饰符] 转换字符

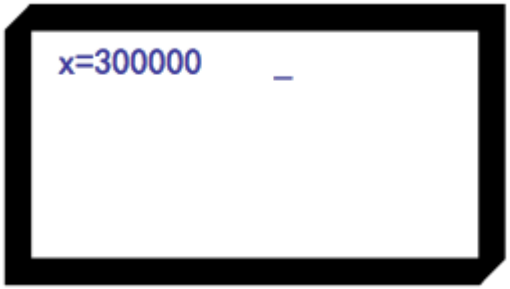
-、0等， 例3.6

输出数据类型和格式
表3-1 d, u, c, s, f

h、l、L：指出输出参数的类型
例3.8

m.n: 设置显示的最小宽度及小数位数
例3.6

```
long x=300000;
printf("x=%-10ld ", x);
```



长度修饰符

h：加在 d、o、x、u 之前，表示输出 short

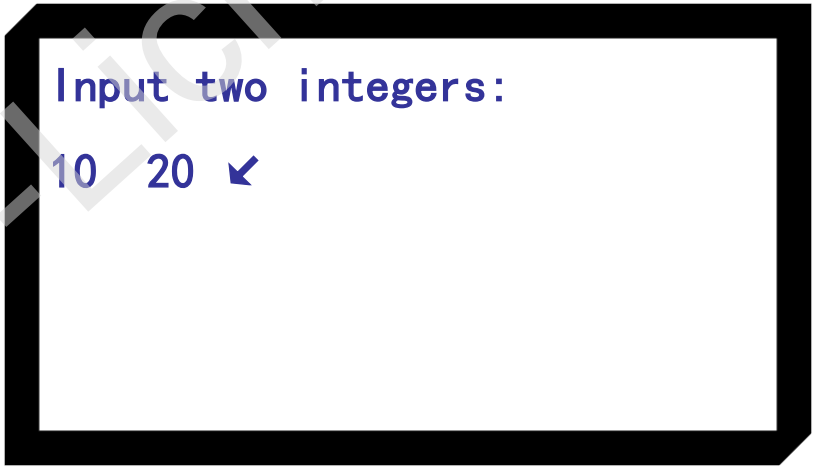
l：加在 d、o、x、u 之前，表示输出 long

L：加在 f、e、g 之前，表示输出 long double

```
short a;
long b;
long double y;
printf("a=%hd,b=%ld,y=%Lf\n",a,b,y);
```

函数scanf

在标准输入设备上按指定格式输入各种类型的数据到内存中。



例：

```
int x, y;
printf ("Input two integers:\n");
scanf("%d %d", &x, &y);     // P77(2)
```

scanf 的调用方式

scanf (“•••” ，地址参数1，地址参数2，•••) ;

格式控制串

1. 格式说明（表3-2）

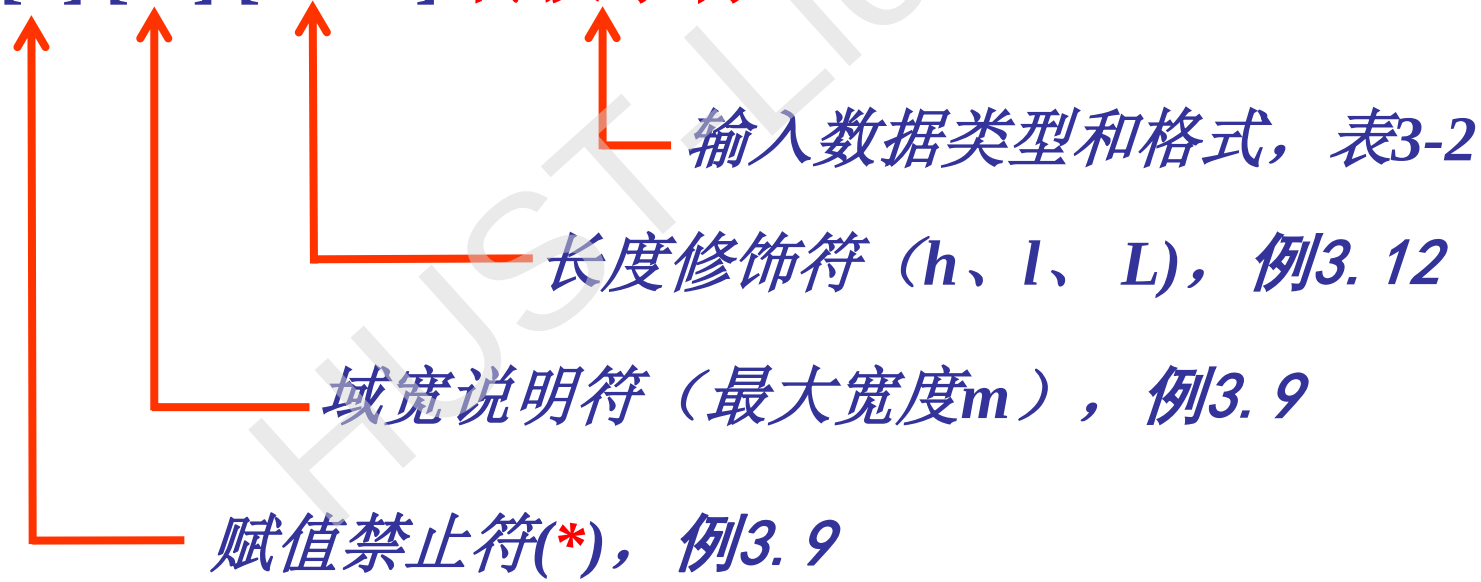
2. 空白字符：空格，\n , \t（被忽略）

3. 非空白字符：在输入流的相应位置必须有相同的字符与之匹配（除逗号外，其它最好不要用）

格式说明的基本组成

scanf (“•••” ，地址参数1，地址参数2，•••) ;

% [*] [m] [h/l/L] 转换字符



长度修饰符

```
long a;
double x;
scanf(“ %ld  %lf ”, &a , &x) ;
```

//注意：输入double型数据用%lf，而非%f

转换说明与输入参数类型不匹配时可能导致的后果：

读入的数据值不正确或程序非正常终止；（P77(2)）

输入实数时不能规定精度　P78(4)

如 float a;

```
scanf(“%5.2f”,&a);
```

//错误，无论输入什么数，a=0.000000

输入数据的格式（数据项的分隔）

1. 用隐含的分隔符（空格、回车键）（例3.9、3.11）

   （用隐含的分隔符分隔整数、浮点数、字符串）

2. 根据转换字符的含义从输入流中取得数据（例3.10）

   （字符不需分隔，根据转换字符的含义自动分隔）

3. 根据转换项中指定的域宽分隔出数据项（例3.9）

4. 使用显示的分隔符（P78（3））

   如逗号’,’ 在输入流的相应位置必须有相同的字符与之一一对应，否则易出错

隐含的分隔符

```
#include<stdio.h>
int main()
{
  int a,b;
  char op;
  printf(“Input two integers  \n");
  scanf("%d%d", &a, &b);
  printf("Input an operator\n");
  scanf("%c", &op);
  printf("%d%c%d\n ", a, op, b);
  return 0;
}
```

```
Input two integers
2   3↙
Input an operator
2
3
```

赋值禁止符 *

```
#include<stdio.h>
int main()
{
  int a,b;
  char op;
  printf("Input two integers \n ");
  scanf("%d%d", &a, &b);
  printf("Input an operator\n");
  scanf("%*c%c", &op);
  printf("%d%c%d\n ", a, op, b);
  return 0;
}
```

```
Input two integers
2   3↙
Input an operator
+↙
2 + 3
```

*：表示跳过相应的数据，不赋值

显示分隔符

```
#include<stdio.h>
int main()
{
  int a,b;
  char op;
  printf("Input two integers, separated by Commas\n");
  scanf("%d ,%d", &a, &b);
  printf("Input an operator:");
  scanf("%*c%c", &op);
  printf("%d %c %d \n ", a, op, b);
  return 0;
}
```

```
Input two integers, separated by Commas
2 , 3↙
Input an operator:
+ ↙
2 +3
```

显示分隔符，即用户定义的分隔符（要有提示）
如按点分十进制输入IP地址

例

求两个整数之和的程序

```c
#include <stdio.h>      //预处理

int main()              //函数定义
{
    int a=0, b=0;           //变量说明
    int sum;

    printf("please input a and b using (a b):\n");    //提示输入信息
    scanf("%d %d",&a,&b);  //数据输入
空格分隔 ——↑
    sum = a+b;      //执行任务

    printf("sum=%d\n",sum);

    return 0;           //函数返回
}
```

```
please input a and b using (a b):
3 4
sum=7
```

```
please input a and b using (a b):
3
4
sum=7
```

```
please input a and b using (a b):
3,4
sum=3
```