一、 Printf 输出格式   
C中格式字符串的一般形式为： %[标志][输出最小宽度][.精度][长度]类型，

1.类型类型字符用以表示输出数据的类型，其格式符和意义下表所示：

| **表示输出类型的格式字符** | **格式字符意义** |
| --- | --- |
| a | 浮点数、十六进制数字和p-计数法(C99) |
| A | 浮点数、十六进制数字和p-计数法(C99) |
| c | 输出单个字符 |
| d | 以十进制形式输出带符号整数(正数不输出符号) |
| e | 以指数形式输出单、双精度实数 |
| E | 以指数形式输出单、双精度实数 |
| f | 以小数形式输出单、双精度实数 |
| g | 以%f%e中较短的输出宽度输出单、双精度实数,%e格式在指数小于-4或者大 于等于精度时使用 |
| G | 以%f%e中较短的输出宽度输出单、双精度实 |
| i | 有符号十进制整数(与%d相同) |
| o | 以八进制形式输出无符号整数(不输出前缀O) |
| p | 指针 |
| s | 输出字符串 |
| x | 以十六进制形式输出无符号整数(不输出前缀OX) |
| X | 以十六进制形式输出无符号整数(不输出前缀OX) |
| u | 以十进制形式输出无符号整数 |

标志字符为-、+、#、空格和0五种，其意义下表所示：

| **标志格式字符** | **标 志 意 义** |
| --- | --- |
| - | 结果左对齐，右边填空格 |
| + | 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 |
| # | 对c，s，d，u类无影响；对o类，在输出时加前缀0；对x类，在输出时加前缀0x或者0X；对g，G 类防止尾随0被删除；对于所有的浮点形式，#保证了即使不跟任何数字，也打印一个小数点字符0 |
| 0 | 对于所有的数字格式，用前导0填充字段宽度，若出现-标志或者指定了精度(对于整数),忽略 |

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，若实际位数少于定义的宽度则补以空格或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：如果输出数字，则表示小数的位数；如果输出的是字符，则表示输出字符的个数；若实际位数大于所定义的精度数，则截去超过的部分。

5.长度

* 长度格式符为h,l两种，h表示按短整型量输出，l表示按长整型量输出。   
  h和整数转换说明符一起使用，表示一个short int 或者unsigned short int类型的数值 ，示例：%hu，%hx，%6.4hd
* hh和整数转换说明符一起使用，表示一个short int 或者unsigned short类型的数值 ，示例：%hhu，%hhx，%6.4hhd
* j和整数转换说明符一起使用，表示一个intmax\_t或者uintmax\_t类型的数值 ，示例：%jd,%8jx
* l和整数转换说明符一起使用，表示一个long int 或者unsigned long int类型的数值 ，示例：%ld,%8lu
* ll和整数转换说明符一起使用，表示一个long int 或者unsigned long int类型的数值 (C99)，示例：%lld,%8llu
* L和浮点转换说明符一起使用，表示一个long double的值，示例%Lf，.4Le
* t和整数转换说明符一起使用，表示一个ptrdiff\_t值(两个指针之间的差相对应的类型)(C99)，示例：%td,ti
* z和整数转换说明符一起使用，表示一个size\_t值(sizeof返回的类型)(C99)，示例：%zd,zx

/\*使用printf函数时还要注意一个问题，那就是输出表列中的求值顺序。不同的编译系统不一定相同，可以从左到右，也可从右到左。Turbo C是按从右到左进行的\*/

main()

{

int i=8;

printf("%d\n%d\n%d\n%d\n%d\n",++i,--i,i--,i++,-i--)

}

6.特殊用法

char ch[20];   
printf(“%\*.\*s\n”,m,n,ch);   
前边的\*定义的是总的宽度，后边的定义的是输出的个数。分别对应外面的参数m和n 。

看到一种输出格式 %n 可以将所输出字符串的长度值赋绐一个变量,

int n   
printf(“hello world%n”, &n);   
执行后变量被赋值为11。

二、c++ cout 输出格式(头文件iomanip)  
　　io代表输入输出，manip是manipulator（操纵器）的缩写   
　　iomanip的作用: 主要是对cin,cout之类的一些操纵运算子，setfill,setw,setbase,setprecision等等。它是I/O流控制头文件,就像C里面的格式化输出一样.以下是一些常见的控制函数的:   
　　dec 置基数为10 相当于”%d”   
　　hex 置基数为16 相当于”%X”   
　　oct 置基数为8 相当于”%o”   
　　setfill( ‘c’ ) 设填充字符为c   
　　setprecision( n ) 设显示小数精度为n位   
　　setw( n ) 设域宽为n个字符   
　　这个控制符的意思是保证输出宽度为n。如：   
　　cout << setw( 3 ) << 1 << setw( 3 ) << 10 << setw( 3 ) << 100 << endl; 输出结果为   
　　1 10100 （默认是右对齐）当输出长度大于3时(<<1000)，setw(3)不起作用。 