

数据输入输出

C++语言中没有提供专门的输入输出语句，所有的输入输出都是调用标准库函数中的输入输出函数来实现的。在使用时，应在源程序的开头使用如下语句：

```
#include<iostream>
#include<cstdio>
using namespace std;
```

C++语言标准函数库提供了许多标准输入、输出函数，本节将介绍6个最基本的输入、输出函数：

1. 字符输入getchar()
2. 字符输出putchar()
3. 格式化输入scanf()
4. 格式化输出printf()
5. 流输入cin
6. 流输出cout

一、字符输入函数getchar

getchar函数是接收从键盘输入的单个字符数据。它是一个无参函数，其语法格式为：`getchar();`

【说明】

1)通常把输入的字符赋予一个字符变量，构成赋值语句。例如：

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

2) getchar函数只能接受单个字符，输入数字也按字符处理。

3)输入多于一个字符时，只接收第一个字符。

4)getchar函数等待用户输入，直到按回车键才结束，可用于暂停程序的运行，直到输入一个回车键。

5)如果在程序中连续有两个以上getchar()函数，应该一次性输入所需字符，最后再按回车键，否则会把回车作为一个字符传给后面的getchar()函数。

例2.8 利用getchar函数接收键盘输入。

```
#include<cstdio>
#include<iostream>
using namespace std;
int main()
{
    char ch;
    ch=getchar();           //读入字符
    cout<<"input="<<ch<<endl;
}
```

二、字符输出函数putchar

putchar函数是字符输出函数，功能是向标准输出设备(如显示器)输出单个字符数据，其语法格式为：

```
putchar(ch); //其中，ch为一个字符变量或常量。
```

例2.9 利用putchar 函数输出字符。

```
#include<cstdio>
#include<iostream>
using namespace std;
int main()
{
    char c='B';           //定义字符变量c并赋值'B'
    putchar(c);          //输出该字符
    putchar('\x42');      //用转义字符输出字母'B'
    putchar(0x42);        //用16进制ASCII码值输出字母'B'
    putchar(66);          //用10进制ASCII码值输出字母'B'
}
```

运行结果：BBBB

三、通过cout流输出数据

流插入运算符<<和cout结合在一起使用，可向显示器屏幕输出数据。

格式1：

功能：它把表达式的值输出到屏幕上，该表达式可以是各种基本类型的常量、变量或者由它们组成的表达式。输出时，程序根据表达式的类型和数值大小，采用不同的默认格式输出，大多数情况下可满足要求。

若要输出多个数据，可以连续使用流插入运算符

格式2：

功能：将表达式的内容一项接一项的输出到屏幕上。

1. 输出字符串和输出变量的区别

每当我们输出字符串常量的时候，必须用双引号把字符串引起来，以便将它和变量名明显的区分开来。

例2.10

下面两个语句是不同的：

```
cout << "Hello"; //打印字符串Hello到屏幕上cout << Hello ;
cout << Hello;   //把变量Hello存储的内容打印到屏幕上
```

2. 如何增强信息的可读性

为了增强输出信息的可读性，在输出多个数据时可以通过插入空格符，换行符或其他提示信息将数据进行组织，以获得更好的效果。

例2.11

```
x=12;
cout << "tom is my friend,he is";
cout << x;
cout <<"years old";
```

输出结果为tom is my friend,he is12year old

3. 换行符的使用

必须注意，除非我们明确指定，cout并不会自动在其输出内容的末尾加换行符，因此下面的语句：

例2.12

```
cout << "This is a sentence." ;  
cout << "This is another sentence." ;
```

将会有如下内容输出到屏幕：

```
This is a sentence.This is another sentence.
```

虽然我们分别调用了两次cout，两个句子还是被输出在同一行。所以，为了在输出中换行，我们必须插入一个换行符来明确表达这一要求，在C++中换行符可以写作\n。

```
cout << "First sentence.\n";  
cout << "Second sentence.\nThird sentence.";
```

将会产生如下输出：

```
First sentence.  
Second sentence.  
Third sentence.
```

例2.13

在屏幕上输出

```
2 3
```

```
4
```

```
cout<<2<<" "<<3<<endl;
```

```
cout<<4;
```

```
或 cout<<"2 3\n4";
```

除了以上两种写法外，还可以有其它的写法，请试试看。你可以使用\n或endl来指定cout输出换行，注意两者的不同用法。

四、通过cin流读入数据

流读取运算符>>和cin结合在一起使用，可从键盘输入数据。

格式1：

功能：是从键盘读取一个数据并将其赋给“变量”。

说明：在使用cin输入的时候必须考虑后面的变量类型。如果你要求输入一个整数，在 >>后面必须跟一个整型变量，如果要求一个字符，后面必须跟一个字符型变量。

例2.14声明一个整型变量age然后等待用户从键盘输入到cin并将输入值存储在这个变量中。

```
int age;
```

```
cin >> age;
```

也可以连续使用>>，实现从键盘对多个变量输入数据。

格式2：

这要求从键盘输入的数据的个数、类型与变量相一致。从键盘读取数据时，各数据之间要有分隔符，分隔符可以是一个或多个空格键、回车键等。

例2.15用cin 让用户输入多个数据 。

```
cin >> a >> b;
```

等同于：

```
cin >> a;
```

```
cin >> b;
```

例2.16流读取运算符>>和cin的使用

```
#include<iostream>
using namespace std;
int main( )
{ char c;
  int i;
  float x,y;
  cout<<"enter:\n";
  cin >>i>>x>>y;
  c=i;
  cout<<"c="<<c<<"\t i="<<i<<"\n";
  cout<<"x="<<x<<"\t y="<<y<<"\n";
  return 0;
}
```

程序运行时屏幕先显示:

enter:

这时从键盘输入一个整数和两个实数,中间用一个或多个空格键作分隔符。如输入

65 2.3 3.5

最后屏幕显示:

c=A i=65

x=2.3 y=3.5

程序中"\n"和"\t"都是转义符。'\n'和"\n"效果相同,都表示换行符。'\t'是制表符,可以理解为连续输出几个空格。字符变量和整型变量i的值都是65,但输出的形式不同。

五、格式化输入函数scanf

scanf函数的功能是格式化输入任意数据列表,其一般调用格式为:

scanf(格式控制符,地址列表)

【说明】

- 1) 地址列表中给出各变量的地址,可以为变量的地址,也可以为字符串的首地址。
- 2) 格式控制符由%和格式符组成,作用是将要输入的字符按指定的格式输入,如%d,%c等。

表2-6 scanf函数的格式符

格式符	说明
d, i	用于输入十进制整数
u	以无符号十进制形式输入十进制整数
o(字母)	用于输入八进制整数
x	用于输入十六进制整数
c	用于输入单个字符
s	用于输入字符串(非空格开始,空格结束,字符串变量以'\0'结尾)
f	用于输入实数(小数或指数均可)
e	与f相同(可与f互换)

表2-7 scanf函数的附加格式说明符

附加格式	说明
l(字母)	用于长整型数(%ld,%lo,%lx)或double型实数(%lf,%le)
h	用于短整型数(%hd,%ho,%hx)

域宽(一个整数)	指定输入所占列宽
*	表示对应输入量不赋给一个变量

```

例.1 *格式符的使用
#include<stdio>
int main()
{
    int a,b;
    scanf("%d*d*d",&a,&b);
    printf("a=%d,b=%d\n",a,b);
}
输入: 1 2 3回车
输出: a=1,b=3

```

```

例.2 域宽格式符的使用
#include<stdio>
int main()
{
    int a,b;
    scanf("%4d%4d",&a,&b);
    printf("a=%d,b=%d\n",a,b);
}
输入: 1234567回车
输出: a=1234,b=567

```

```

例.3 %s格式符的使用
#include<stdio>
int main()
{
    char st[40];
    scanf("%s",st);
    printf("Your input is:%s\n",st);
}
输入: abc hello
输出: abc

```

```

例.4 非格式符的使用
#include<stdio>
int main()
{
    int a,b,c;
    scanf("%d,%d,%d",&a,&b,&c);
    printf("a=%d,b=%d,c=%d\n",a,b,c);
} //例中scanf用非格式字符','作间隔符
故输入应为: 5,6,7 (本来输入:5 6 7)
输出: a=5,b=6,c=7

```

28

六、格式化输出函数printf

printf函数的功能是格式化输出任意数据列表，其一般调用格式为：

printf(格式控制符，输出列表)

【说明】

- 格式控制由输入格式说明和普通字符组成，必须用双引号括起来。
 - 格式说明由%和格式字符组成，作用是将要输出的字符转换为指定的格式，如%d，%c等。
 - 普通字符是在输出时原样输出的字符，一般在显示时起提示作用。
- 输出列表是需要输出的一组数据（可以为表达式和变量），各参数之间用“，”分开。要求格式说明和各输出项在数量和类型上要一一对应，否则将会出现意想不到的错误。

表2-8 printf函数的格式符

格式符	说 明
d(或i)	以带符号的十进制形式输出整数，正数的(+)号省略不输出
u	以无符号十进制形式输出整数
x(或X)	以十六进制无符号形式输出整数（不输出前导符0x）
o(字母)	以八进制无符号形式输出整数（不输出前导符数字0）
c	输出一个字符
s	输出字符串
f	以小数形式输出单、双精度，隐含输出6位小数
e(或E)	以指数形式输出单、双精度，隐含输出6位小数
g(或G)	自动选用%f、%e或%E格式中输出宽度较小的一种使用

表2-9 d格式符

参 数	说 明
%d	输出数字长为变量数值的实际长度
%md	输出m位(不足补空格，大于m位时按实际长度输出)
%-md	m含义同上。左对齐输出
%ld	l(小写字母)表示输出“长整型”数据
%mld	指定长整型输出宽度m位，左边补空格；否则，按实际位数输出
%0md,%0mld	0(数字0)表示位数不足m时补0

例如，对如下程序段

```
int i=1;
```

```
long j=123;
printf("%d,%2d,%03d,%1d,%-4d,%05ld",i,i,j,j);
输出: 1, 1,001,123,123 ,00123
```

表2-10 f格式符

参 数	说 明
%f	按实数格式输出，整数部分按实际位数输出，6位小数
%m.nf	总位数m（含小数点），其中有n位小数
%-m.nf	同上，左对齐

表2-11 s格式符

参 数	说 明
%s	按实际宽度输出一个字符串
%ms	m指定宽度（不足时左补空格，大于时按实际宽度输出）
%-ms	左对齐，不足时右补空格
%m.n s	输出占m个字符位置，其中字符数最多n个，左补空格
%-m.n s	同上，右补空格

七、几种输入输出格式的几点说明

1、cin和cout在Dev C++中只能调用<iostream>库，而其他输入输出格式要调用<stdio.h>库或<cstdio>库。

2、cin和cout属于C++的概念，调用时涉及输入输出流，而scanf和printf属于C的概念，是C语言的标准输入/输出库中的函数，所以在时效上，scanf和printf优于cin、cout，对于大数据的输入输出，通常情况下应该用scanf、printf。

3、对于普通数据的输入输出，cin和cout比较方便，而在格式化方面，scanf和printf比较容易。cin效率比scanf低很多，尤其输入数据达到20万以上时非常明显，在100万时cin读入就会超时。

4、scanf和printf也缺点，cin和cout能够自动识别变量的数据类型，因此，在进行输入输出时，不需要指定数据类型，printf和scanf函数在输入输出时需指定数据类型。