流输入输出

C++语言提供了标准的数据输入、输出方法，cin 和 cout 就是最常用的方式。

使用 cin 和 cout 需要:

```
1. #include<iostream>
```

通过 cin 流读入数据

流提取运算符>>和 cin 结合在一起使用，可从键盘输入数据。

【对一个变量输入数据】

从键盘读取一个数据并将其赋给“变量”。格式如下：

```
cin >> 变量名;
```

例如，声明一个整型变量 a，等待用户从键盘输入并将输入值存储在变量 a 中。

```
int a;
cin >> a;
```

【对多个变量输入数据】

连续使用>>，实现从键盘对多个变量输入数据。格式如下：

```
cin >> 变量名 >> 变量名;
```

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。

例如，用 cin 让用户输入多个数据。

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

等同于：

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

通过 cout 流输出数据

流插入运算符<<和 cout 结合在一起使用，可向屏幕输出数据。

【输出一项数据】

向屏幕输出一项数据。格式如下：

```
1.  cout << 输出内容;
```

把表达式的值输出到屏幕上，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。

输出字符串常量的时候，必须用双引号把字符串引起来，以便和变量名区分开来。

下面两个语句是不同的：

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

【输出多项数据】

若要输出多个数据，可以连续使用流插入运算符。

```
1.  cout << 输出内容 << 输出内容;
```

【换行符的使用】

cout 并不会自动在输出内容的末尾加换行符。

```
1.  cout << "First sentence.";
2.  cout << "Second sentence.";
```

上面的代码虽然调用了两次 cout，但是两个句子输出在同一行。输出结果为：

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

在输出中换行，必须明确表达这一要求。

可以用操作符 endl 来换行：

```
1.  cout << "First sentence." << endl;
2.  cout << "Second sentence." << endl;
```

上面程序的输出结果为：

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

换行也可以写作\n：

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

上面程序的输出结果为：

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

多种多样的输出

【问题描述】

用多种方法在屏幕上输出如下内容:

```
2 3
4
```

【不同的方案】

```
cout << "2 3" << endl;
cout << "4";

cout << "2 3" << endl << "4";

cout << 2 << " " << '3' << endl;
cout << "4";
```

流输出的格式化控制

我们可以对 cout 输出的内容进行格式化控制，前提是包含如下的头文件：

```
#include<iomanip>
```

【参考程序】

```
#include<iostream>
#include<iomanip>
using namespace std;
int main()
{
    //输出宽度 5
    cout << setw(5) << 513 << endl;
    //输出宽度 5，其余位置用 0 补齐
    cout << setfill('0') << setw(5) << 255 <<endl;
    //总共按照 3 位输出
    cout << setprecision(3) << 3.1415926 << endl;
    //小数点后保留 3 位
    cout << fixed << setprecision(3) << 3.1415926 << endl;
    return 0;
}
```

【输出结果】

```
  513
00255
3.14
3.142
```

按照要求输出

【问题描述】

输入一个整数和一个浮点数。

第一行按照 5 位宽度输出整数，整数右对齐。

第二行按照 5 位宽度输出整数，整数右对齐，前面补 0。

第三行输出浮点数，总位数输出 3 位。

第四行输出浮点数，小数点后位数输出 3 位。

【输入样例】

```
255 3.14159
```

【输出样例】

```
  255
00255
3.14
3.142
```

【参考代码】

```
#include<iostream>
#include<iomanip>
using namespace std;
int main()
{
    int a;
    double b;
    cin >> a >> b;
    //输出宽度 5
    cout << setw(5) << a << endl;
    //输出宽度 5，其余位置用 0 补齐
    cout << setfill('0') << setw(5) << a <<endl;
    //总共按照 3 位输出
    cout << setprecision(3) << b << endl;
    //小数点后保留 3 位
    cout << fixed << setprecision(3) << b << endl;
```

```
    return 0;
}
```