



北京大学  
PEKING UNIVERSITY

信息科学技术学院

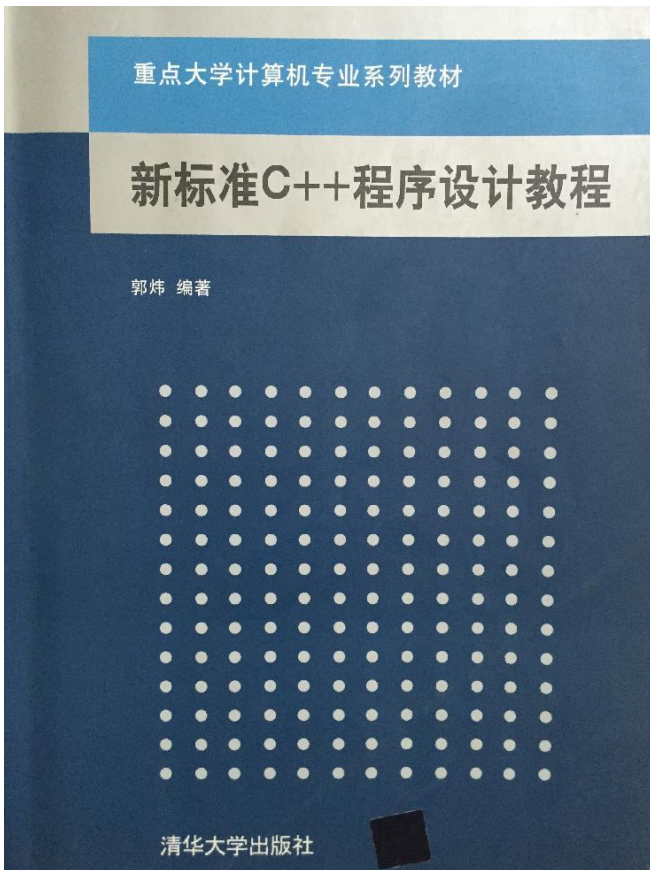
指定教材:

# 程序设计与算法(一)

李文新 郭炜

主讲教师互动微博:

<http://weibo.com/guoweiofpku>



# 输入输出控制符

在printf和scanf中可以使用以“%”开头的控制符，指明要输入或输出的数据的类型以及格式。

```
int n = 3;  
printf("I want to buy %d books for %f dollars",3,4.5);  
double f;  
scanf("%d%f",&n,&f);
```

# 输入输出控制符

在printf和scanf中可以使用以“%”开头的控制符，指明要输入或输出的数据的类型以及格式。

常用格式控制符	作用
%d	读入或输出int变量
%c	读入或输出char变量
%f	读入或输出float变量，输出时保留小数点后面6位
%lf	读入或输出double变量，输出时保留小数点后面6位
%x	以十六进制读入或输出整型变量
%lld	读入或输出long long 变量(64位整数)
%nd（如%4d,%12d）	以n字符宽度输出整数，宽度不足时用空格填充
%0nd（如%04d,%012d）	以n字符宽度输出整数，宽度不足时用0填充
%.nf（如%.4f,%3f）	输出double或float值，精确到小数点后n位

# 用scanf读入不同类型的变量

用scanf可以一次读入多个类型不同的变量，只要输入的各项之间用空格分隔即可。

```
#include <iostream>
#include <cstdio> //使用printf,scanf要有这一行
using namespace std;
int main()
{
    int n; char c; float m;
    scanf("%d%c%f", &n, &c, &m);
    printf("%d %c %f", n, c, m);
    return 0;
}
```

● 依次输入一个整数，一个字符，再一个整数，则它们会被分别放入n, c, m

34k 234.45 ✓

34 k 234.449997

# 用scanf读入不同类型的变量

用scanf可以一次读入多个类型不同的变量，只要输入的各项之间用空格分隔即可。

```
#include <iostream>
#include <cstdio>
using namespace std;
int main()
{
    int n; char c; float m;
    scanf("%d%c%f", &n, &c, &m);
    printf("%d %c %f", n, c, m);
    return 0;
}
```

- 依次输入一个整数，一个字符，再一个整数，则它们会被分别放入n, c, m
- **&n**代表“取n的地址”

34k 234.45 ✓

34 k 234.449997

# 用scanf读入不同类型的变量

用scanf可以一次读入多个类型不同的变量，只要输入的各项之间用空格分隔即可。

```
#include <iostream>
#include <cstdio>
using namespace std;
int main()
{
    int n; char c; float m;
    scanf("%d%c%f", &n, &c, &m);
    printf("%d %c %f", n, c, m);
    return 0;
}
```

- 依次输入一个整数，一个字符，再一个整数，则它们会被分别放入n, c, m
- **&n**代表“取n的地址”
- **%c**代表等待输入一个字符

34k 234.45 ✓

34 k 234.449997

# 用scanf读入不同类型的变量

用scanf可以一次读入多个类型不同的变量，只要输入的各项之间用空格分隔即可。

```
#include <cstdio>
#include <iostream>
using namespace std;
int main()
{
    int n; char c; float f;
    scanf("%d%c%f", &n, &c, &f);
    printf("%d %c %f", n, c, f);
    return 0;
}
```

- 依次输入一个整数，一个字符，再一个整数，则它们会被分别放入n, c, m
- **&n**代表“取n的地址”
- **%c**代表等待输入一个字符
- **%f**代表等待输入一个**float**小数

<u>34k 234.45</u> ✓ 34 k 234.449997
--

**%f**用于输出**double**或**float**类型的值，保留小数点后面**6**位

# 用scanf读入不同类型的变量

输入字符时，不会跳过空格(空格也会被当作字符读入)，  
输入其他类型的数据时，会跳过空格

```
#include <iostream>
#include <cstdio>
using namespace std;
int main()
{
    int n; char c; float f;
    scanf("%d%c%f", &n, &c, &f);
    printf("%d %c %f", n, c, f);
    return 0;
}
```

34 k 456 ✓
34 0.000000

c = ' ', 读入f时，对应输入是'k'，导致出错



# 用scanf跳过输入中的字符

如果在输入中有scanf中出现的非控制字符，  
则这些字符会被跳过

```
#include <iostream>
#include <cstdio>
using namespace std;
int main()
{
    int n,m; char c; float f;
    scanf("%d %c,%f:%d",&n,&c,&f,&m);
    printf("%d,%c,%f,%d",n,c,f,m);
    return 0;
}
```

12 k,3.75:290 ✓  
12,k,3.750000,290

## 控制printf 输出整数的宽度

<code>%nd</code> (如 <code>%4d,%12d</code> )	以n字符宽度输出整数, 宽度不足时用空格填充
<code>%0nd</code> (如 <code>%04d,%012d</code> )	以n字符宽度输出整数, 宽度不足时用0填充

```
int n = 123;  
printf("%05d,%5d,%5d,%05d",n,n,123456,123456);
```

00123, 123,123456,123456

## 控制printf 输出浮点数的精度

<code>%.nf</code> (如 <code>%.4f</code> , <code>%.3f</code> )	输出浮点数, 精确到小数点后n位
--	------------------

```
float a = 123.45;  
double b = 22.37362723;  
printf("%.5f %.2f %.12f", 12.3, a, b);
```

12.30000 123.45 22.373627230000
---------------------------------

# 尽量使用double

- double 精度高于float, 所以一般尽量使用double!
- double类型的变量用 **%lf** 输入!

```
int main()
{
    float f,c;
    scanf("%f",&f);
    c = 5 * (f-32)/9;
    printf("%.5f",c,f);
    return 0;
}
```

输入: 1501

输出: 816.11108

```
int main()
{
    double f,c;
    scanf("%lf",&f);
    c = 5 * (f-32)/9;
    printf("%.5f",c,f);
    return 0;
}
```

输入: 1501

输出: 816.11111

使用 double 的结果更精确!!!!

## 格式控制符%x和%u

- **%x**: 以十六进制形式读入或输出整数
- **%u**: 以无符号整数形式输出整数

```
printf ("%x,%d,%u", 0xffffffff, 0xffffffff, 0xffffffff) ;  
=> ffffffff,-1,4294967295
```

# 格式控制符%x和%u

- **%x**: 以十六进制形式读入或输出整数
- **%u**: 以无符号整数形式输出整数

```
printf ("%x,%d,%u", 0xffffffff, 0xffffffff, 0xffffffff);  
=> ffffffff,-1,4294967295
```

```
int n;  
scanf ("%x", &n);  
printf ("%d", n);
```

12 ✓

18

a0 ✓

160

# 用C++的cout进行输出

```
#include <iostream>
using namespace std;
int main()    {
    int n=5;
    double f = 3.9;
    char c = 'a';
    cout << "n=" << n << ",f=" << f << endl; //endl换行
    cout << 123 << ", c=" << c;
    return 0;
}
```

```
n=5,f=3.9
123, c=a
```

# 用C++的cin进行输入

```
#include <iostream>
using namespace std;
int main()    {
    int n1,n2;
    double f;
    char c;
    cin >> n1 >> n2 >> c >> f;
    cout << n1 << "," << n2 << "," << c << "," << f;
    return 0;
}
```

5 10k 1.23 ✓  
5,10,k,1.23



## 用cin读入所有输入的字符，包括空格，回车

```
#include <iostream>
using namespace std;
int main()
{
    int c;
    while((c = cin.get()) != EOF) {
        cout << (char)c ;
    }
    return 0;
}
```

## 用scanf读入所有输入的字符，包括空格，回车

```
#include <iostream>
#include <cstdio>
using namespace std;
int main()
{
    char c;
    while (scanf("%c",&c) != EOF) {
        printf("%c",c);
    }
    return 0;
}
```

## 用cin,cout和scanf,printf

- cin,cout 速度比scanf,printf慢,输入输出数据量大时用后者
- 一个程序里面不要同时用cin和scanf,不要同时用cout和printf