

C/C++ 语言

循环

张晓平

武汉大学数学与统计学院

1. 使用函数返回值的循环

使用函数返回值的循环

使用函数返回值的循环

例

编制一个函数，计算一个数的整数次幂，如 n^p 。

使用函数返回值的循环

例

编制一个函数，计算一个数的整数次幂，如 n^p 。

请大家记住，头文件 `math.h` 中提供了一个名为 `pow` 的幂函数，允许计算浮点数次幂。

pow 函数的使用

```
1 // pow.c:
2 #include <stdio.h>
3 #include <math.h>
4 int main(void)
5 {
6     printf("    2^3    = %f\n", pow(2.0,3.0));
7     printf("  sqrt 2  = %f\n", pow(2.0,0.5));
8     printf("  3^(1/4) = %f\n", pow(3.0,0.25));
9     return 0;
10 }
```

pow 函数的使用

```
2^3    = 8.000000  
sqrt 2  = 1.414214  
3^(1/4) = 1.316074
```

使用函数返回值的循环

```
1 double power(double n, int p)
2 {
3     double pow = 1;
4     int i;
5     for (i = 1; i <= p; i++)
6         pow *= n;
7     return pow;
8 }
```


使用函数返回值的循环

写一个具有返回值的函数需要做以下事情：

- 定义函数时，说明它的返回值类型；
- 使用关键字 `return` 指示要返回的值。

使用函数返回值的循环

- 要声明函数类型，可以在函数名之前写出类型，就像声明一个变量时那样；
- 关键字 `return` 使函数把它后面的值返回给调用函数。可以返回一个值，也可以返回一个表达式。

```
return 2 * x + b;
```

使用函数返回值的循环

在调用函数中，

- 可以把一个返回值赋给另一个变量。

```
b = power(1.2, 3);
```

- 可以把一个返回值作为一个表达式中的值。

```
b = 2.0 + power(1.2, 3);
```

- 也可以把一个返回值作为另一个函数的参数。

```
printf("%f", power(1.2, 3));
```

power 函数的测试 i

```
// power.c:
#include <stdio.h>
double power(double n, int p);
int main(void)
{
    double x, xpow;
    int exp;
    printf("Enter a number and the positive\ninteger power\n");
    printf("Enter q to quit.\n");
    while (scanf("%lf%d", &x, &exp) == 2) {
        xpow = power(x, exp);
        printf("%.3g to the power %d is %.5g\n",
            x, exp, xpow);
    }
}
```

power 函数的测试 ii

```
    printf("Enter next pair of numbers or q to  
    quit.\n");  
}  
printf("Hope you enjoyed this power trip --  
bye!\n");  
return 0;  
}  
double power(double n, int p)  
{  
    double pow = 1;  
    int i;  
    for (i = 1; i <= p; i++)  
        pow *= n;  
    return pow;  
}
```


power 函数的测试

```
Enter a number and the positive integer power
to which
the number will be raised. Enter q to quit.
1.2 12
1.2 to the power 12 is 8.9161
Enter next pair of numbers or q to quit.
2
16
2 to the power 16 is 65536
Enter next pair of numbers or q to quit.
q
```