



阶乘之和

求： $1! + 2! + 3! + \dots + 10!$

如果 C++ 提供了计算阶乘的函数 `factorial(x)` 的话，我们就可以用如下的方式来编写程序：

```
1. #include<iostream>
2. using namespace std;
3. int main()
4. {
5.     int sum = 0;
6.     for (int i=1; i<=10; i++)
7.     {
8.         sum += factorial(i);
9.     }
10.    cout << sum << endl;
11.    return 0;
12. }
```

可是 C++ 没有提供 `factorial(x)` 这样一个标准函数，所以这个程序是通不过的。但是我们可以编写自己的函数。

我们已经知道，如果是 C++ 的标准函数，只需要在程序中通过 `#include` 指令加入相应的库即可以直接调用，比如 `abs(x)`，`sqrt(x)` 等。

下面我们就来学习如何编写自己的函数。

```
1. //函数名 factorial, 形参 int n
2. int factorial(int n)
3. {
4.     int p = 1;
5.     for (int i=1; i<=n; i++)
6.     {
7.         p *= i;
8.     }
9.     return p;
10. }
```

本例中，函数名叫 `factorial`，只有一个 `int` 型的参数 `n`，函数 `factorial` 的返回值是 `int` 型。



函数中，用到两个变量 i , p 。在函数体中，是一个求阶乘的语句， n 的阶乘的值在 p 中，最后由 `return` 语句将计算结果 p 值带回，`factorial()`函数执行结束。

在这里，函数的参数 n 是一个形参（形式参数）。

如果我们调用函数：`factorial (3)`，那么在函数内部所有的 n 都被替代成 3 来进行计算。在这里，3 就被称为实参（实际参数）。

完整代码，如下所示。第 9 行 `factorial(i)`就返回了子函数计算出来的的 i 的阶乘的值。

```
1. #include<iostream>
2. using namespace std;
3. int factorial(int n);
4. int main()
5. {
6.     int sum=0;
7.     for (int i=1; i<=10; i++)
8.     {
9.         sum += factorial(i);
10.    }
11.    cout << sum << endl;
12.    return 0;
13. }
14. //函数名 factorial, 形参 int n
15. int factorial(int n)
16. {
17.     int p = 1;
18.     for (int i=1; i<=n; i++)
19.     {
20.         p *= i;
21.     }
22.     return p;
23. }
```