

C语言程序设计(第4章函数)4 PDF转换可能丢失图片或格式，
建议阅读原文

https://www.100test.com/kao_ti2020/134/2021_2022_C_E8_AF_AD_E8_A8_80_E7_A8_8B_c97_134853.htm 4.5 实现问题 在编写C语言的函数时，有几个要点需要我们牢记，因为它们影响到函数的效率和可用性。

4.5.1 参数和通用函数 通用函数是指能够被用在各种情况下，或者是可被许多不同程序员使用的函数。我们不应该把通用函数建立在全局变量上（不应该在通用函数中使用全局变量）。函数所需要的所有数据都应该用参数传递（在个别难以这样做的情况下，可以使用静态变量）。使用参数传递，除了有助于函数能用在多种情况下之外，还能提高函数代码的可读性。不用全局变量，可以使得函数减少因副作用而导致错误的可能性。

4.5.2 效率 函数是C语言的基本构件。对于编写简单程序之外的所有程序来说，函数是必不可少的。但在一些特定的应用中，应当消除函数，而采用内嵌代码。内嵌代码是指一个函数的语句中不含函数调用语句。仅当执行速度是很关键的场合下，才用内嵌代码而不用函数。有两个原因使得内嵌代码的执行速度比函数快。首先，调用需要花费时间；其次，如果有参数需要传递，就要把它们放在堆栈中，这也要用时间。在几乎所有的应用中，执行时间上的这些微小开销是微不足道的。不过当时间开销至关重要时，使用内嵌代码消除函数调用，可以把每次函数调用的开销节省下来。下面的两个程序都是打印从1到10的数的平方。由于函数调用需要花费时间，所以内嵌代码版本运行的比另一个要快。

内嵌函数调用

```
main()
main() { { int
x. int x. for(x=1,x printf ("%d" , x * x ). printf ("%d",sqr( x ) ). } } s q
```

`r(a); int a. { return a*a. }` 100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com