#endif

另一种常见的用法是在如下代码序列中：

```
#if defined (DEBUG) && DEBUG
...
#endif
```

这使得#if 到#endif 之间的语句只有在定义了 DEBUG 而且具有非零值时才被处理。

因为可以使用表达式且因为 0 代表 false，程序员（包括笔者自己）经常使用#if 0 … #endif 这样一对预处理语句包围需要注释的代码块。

12.3.3 #undef 语句

在一些情况下，可能需要使一些已经定义的名称成为未定义的，通过使用#undef 语句就可以这么做。要消除特定名称的定义，编写如下语句：

```
#undef name
```

这样，语句

```
#undef IPAD
```

将消除 IPAD 的定义。之后的#ifdef IPAD 或#if defined(IPAD)语句都将判断为 FALSE。这里总结了关于预处理程序的讨论。

12.4 练习

1. 在机器上找到系统头文件 limits.h 和 float.h。检查这些文件，看看其内容。如果这些文件包含其他头文件，确保跟踪它们并查看它们的内容。
2. 定义一个名为 MIN 的宏，它给出两个值的最小值。然后编写程序来测试这个宏定义。
3. 定义一个名为 MAX3 的宏，它给出 3 个值的最大值。然后编写一个程序来测试这个宏定义。
4. 编写一个名为 IS_UPPER_CASE 的宏，其作用是如果字符是大写字母，就给出非零值。
5. 编写一个名为 IS_ALPHABETIC 的宏，其作用是如果一个字符是字母，

就给出非零值。使用本章定义的 IS_LOWER_CASE 宏和本章练习 4 中定义的 IS_UPPER_CASE 宏。

6. 编写一个名为 IS_DIGIT 的宏，其作用是如果字符是 0～9 的数字，就给出非零值。在另一个名为 IS_SPECIAL 的宏定义中使用它。如果字符是一个特殊字符（也就是说，它既不是字母，也不是数字），IS_SPECIAL 将给出非零结果。一定要使用练习 5 中定义的 IS_ALPHABETIC 宏。

7. 编写一个名为 ABSOLUTE_VALUE 的宏，其作用是计算参数的绝对值。确保能够正确计算 ABSOLUTE_VALUE (x+delta)之类的表达式的值。

```
ABSOLUTE_VALUE (x + delta)
```