

C语言程序设计(第4章函数)6 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_134841.htm 4.7 C语言的预处理程序与注释 C程序的源代码中可包括各种编译指令，这些指令称为预处理命令。虽然它们实际上不是C语言的一部分，但却扩展了C程序设计的环境。本节将介绍如何应用预处理程序和注释简化程序开发过程，并提高程序的可读性。

4.7.1 C语言的预处理程序

ANSI标准定义的C语言预处理程序包括下列命令：`#define` `#error` `#include` `#if` `#else` `#elif` `#endif` `#ifdef` `#ifndef` `#undef` `#line` `#pragma` 非常明显，所有预处理命令均以符号#开头，下面分别加以介绍。

4.7.2 #define 命令

`#define`定义了一个标识符及一个串。在源程序中每次遇到该标识符时，均以定义的串代换它。ANSI标准将标识符定义为宏名，将替换过程称为宏替换。命令的一般形式为：`#define identifier string` 注意，该语句没有分号。在标识符和串之间可以有任意个空格，串一旦开始，仅由一新行结束。例如，如希望TURE取值1，FALSE取值0，可说明两个宏

```
#define TURE 1
#define FALSE 0
```

这使得在源程序中每次遇到TURE或FALSE就用0或1代替。例如，在屏幕上打印“0 1 2”：`printf("%d %d %d", FALSE, TRUE, TRUE)`；宏名定义后，即可成为其它宏名定义中的一部分。例如，下面代码定义了ONE、TWO及THREE的值。

```
#define ONE 1
#define TWO ONE ONE
#define THREE ONE TWO
```

懂得宏替换仅仅是以串代替标识符这点很重要。因此，如果希望定义一个标准错误信息，可编写如下代码：

```
#define E_MS "standard error on input\n"
printf(E_MS);
```

编译程序遇到标识符E_MS时，就用“standard error on input\n”替换。对于编译程序，printf()语句实际是如下形式：
printf("standard error on input\n")；如果在串中含有标识符，
则不进行替换。例如：`#define XYZ this is a test ...`
`printf("XYZ")`；该段不打印"this is a test"而打印"XYZ"。如果串
长于一行，可以在该行末尾用一反斜杠续行，例如：`#define`
`LONG_STRING "this is a very long string that is used as an`
`example"` C语言程序普遍使用大写字母定义标识符。这种约定
可使人读程序时很快发现哪里有宏替换。最好是将所有的
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