

C 语言带参数宏定义

C 语言允许宏带有参数。在宏定义中的参数称为形式参数，在宏调用中的参数称为实际参数。

对带参数的宏，在调用中，不仅要宏展开，而且要用实参去替换形参。

带参宏定义的一般形式为：

```
#define 宏名(形参列表) 字符串
```

在字符串中含有各个形参。

带参宏调用的一般形式为：

```
宏名(实参列表);
```

例如：

```
#define M(y) y*y+3*y //宏定义
// Code
k=M(5); //宏调用
```

在宏调用时，用实参 5 去代替形参 y，经预处理宏展开后的语句为 `k=5*5+3*5`。

【示例】输出两个数中较大的数。

```
1. #include <stdio.h>
2. #define MAX(a,b) (a>b) ? a : b
3. int main() {
4.     int x , y, max;
5.     printf("input two numbers: ");
6.     scanf("%d %d", &x, &y);
7.     max = MAX(x, y);
8.     printf("max=%d\n", max);
9.     return 0;
10. }
```

运行结果：

```
input two numbers: 10 20
```

```
max=20
```

程序第 2 行进行了带参宏定义,用宏名 `MAX` 表示条件表达式 `(a>b)?a:b`,形参 `a`、`b` 均出现在条件表达式中。程序第 7 行 `max=MAX(x, y)` 为宏调用, 实参 `x`、`y`, 将代换形参 `a`、`b`。宏展开后该语句为:

```
max=(x>y) ? x : y;
```

用于计算 `x`、`y` 中的大数。

对带参宏定义的说明

1) 带参宏定义中, 形参之间可以出现空格, 但是宏名和形参列表之间不能有空格出现。例如把:

```
#define MAX(a,b) (a>b)?a:b
```

写为:

```
#define MAX (a,b) (a>b)?a:b
```

将被认为是无参宏定义, 宏名 `MAX` 代表字符串 `(a,b) (a>b)?a:b`。宏展开时, 宏调用语句:

```
max=MAX(x, y);
```

将变为:

```
max=(a,b) (a>b)?a:b(x, y);
```

这显然是错误的。

2) 在带参宏定义中, 形式参数不分配内存单元, 因此不必作类型说明。而宏调用中的实参有具体的值, 要用它们去代换形参, 因此必须作类型说明。这是与函数中的情况不同的。在函数中, 形参和实参是两个不同的量, 各有自己的作用域, 调用时要把实参值赋予形参, 进行“值传递”。而在带参宏中, 只是符号代换, 不存在值传递的问题。

3) 在宏定义中的形参是标识符, 而宏调用中的实参可以是表达式。

【示例】输入 `n`, 输出 `(n+1)^2` 的值。

```
1. #include <stdio.h>
2. #define SQ(y) (y)*(y)
```

```

3. int main() {
4.     int a, sq;
5.     printf("input a number: ");
6.     scanf("%d", &a);
7.     sq = SQ(a+1);
8.     printf("sq=%d\n", sq);
9.     return 0;
10.}

```

运行结果：

```

input a number: 9
sq=100

```

第 2 行为宏定义，形参为 y 。程序第 7 行宏调用中实参为 $a+1$ ，是一个表达式，在宏展开时，用 $a+1$ 代换 y ，再用 $(y)*(y)$ 代换 SQ ，得到如下语句：

```
sq=(a+1)*(a+1);
```

这与函数的调用是不同的，函数调用时要把实参表达式的值求出来再赋予形参，而宏代换中对实参表达式不作计算直接地照原样代换。

4) 在宏定义中，字符串内的形参通常要用括号括起来以避免出错。在上例中的宏定义中 $(y)*(y)$ 表达式的 y 都用括号括起来，因此结果是正确的。如果去掉括号，把程序改为以下形式：

```

1. #include <stdio.h>
2. #define SQ(y) y*y
3. int main() {
4.     int a, sq;
5.     printf("input a number: ");
6.     scanf("%d", &a);
7.     sq = SQ(a+1);
8.     printf("sq=%d\n", sq);
9.     return 0;
10.}

```

运行结果为：

```
input a number: 9
```

```
sq=19
```

同样输入 9，但结果却是不一样的。问题在哪里呢？这是由于替换只作符号替换而不作其它处理而造成的。宏替换后将得到以下语句：

```
sq=a+1*a+1;
```

由于 a 为 9 故 sq 的值为 19。这显然与题意相违，因此参数两边的括号是不能少的。即使在参数两边加括号还是不够的，请看下面程序：

```
1. #include <stdio.h>
2. #define SQ(y) (y)*(y)
3. int main() {
4.     int a, sq;
5.     printf("input a number: ");
6.     scanf("%d", &a);
7.     sq = 200 / SQ(a+1);
8.     printf("sq=%d\n", sq);
9.     return 0;
10. }
```

本程序与前例相比，只把宏调用语句改为：

```
sq=160/SQ(a+1);
```

运行本程序如输入值仍为 9 时，希望结果为 2。但实际运行的结果如下：

```
input a number: 9
```

```
sq=200
```

为什么会得这样的结果呢？分析宏调用语句，在宏代换之后变为：

```
sq=200/(a+1)*(a+1);
```

a 为 9 时，由于“/”和“*”运算符优先级和结合性相同，则先作 200/(9+1)得 20，再作 20*(9+1)最后得 200。为了得到正确答案应在宏定义中的整个字符串外加括号，程序修改如下：

```
1. #include <stdio.h>
2. #define SQ(y) ((y)*(y))
3. int main() {
4.     int a, sq;
```

```
5.    printf("input a number: ");
6.    scanf("%d", &a);
7.    sq = 200 / SQ(a+1);
8.    printf("sq=%d\n", sq);
9.    return 0;
10. }
```

由此可见：对于宏定义不仅应在参数两侧加括号，也应在整个字符串外加括号。