#define 叫做宏定义命令，它也是[C语言](http://c.biancheng.net/c/" \t "http://c.biancheng.net/view/_blank)预处理命令的一种。所谓宏定义，就是用一个标识符来表示一个字符串，如果在后面的代码中出现了该标识符，那么就全部替换成指定的字符串。  
  
我们先通过一个例子来看一下 #define 的用法：

#include <stdio.h>

#define N 100

int main(){

int sum = 20 + N;

printf("%d\n", sum);

**return** 0;

}

运行结果：  
120  
注意第 6 行代码int sum = 20 + N，N被100代替了。  
  
宏定义的一般形式为：

#define  宏名  字符串

#表示这是一条预处理命令，所有的预处理命令都以 # 开头。宏名是标识符的一种，命名规则和变量相同。字符串可以是数字、表达式、if 语句、函数等。

程序中反复使用的表达式就可以使用宏定义，例如：

#define M (n\*n+3\*n)

它的作用是指定标识符M来表示(y\*y+3\*y)这个表达式。在编写代码时，所有出现 (y\*y+3\*y) 的地方都可以用 M 来表示，而对源程序编译时，用 (y\*y+3\*y) 去替换所有的宏名 M，然后再进行编译。  
  
将上面的例子补充完整：

#include <stdio.h>

#define M (n\*n+3\*n)

int main(){

int sum, n;

printf("Input a number: ");

scanf("%d", &n);

sum = 3\*M+4\*M+5\*M;

printf("sum=%d\n", sum);

**return** 0;

}

需要注意的是，在宏定义中表达式(n\*n+3\*n)两边的括号不能少，否则在宏展开以后可能会产生歧义。下面是一个反面的例子：

#difine M n\*n+3\*n

在宏展开后将得到下述语句：

s=3\*n\*n+3\*n+4\*n\*n+3\*n+5\*n\*n+3\*n;

#### 对 #define 用法的几点说明

1) 宏定义是用宏名来表示一个字符串，在宏展开时又以该字符串取代宏名  
2) 宏定义不是说明或语句，在行末不必加分号，如加上分号则连分号也一起替换。  
3) 宏定义必须写在函数之外，其作用域为宏定义命令起到源程序结束。如要终止其作用域可使用#undef命令。例如：

#define PI 3.14159

int main(){

// Code

**return** 0;

}

#undef PI

void func(){

// Code

}

表示 PI 只在 main() 函数中有效，在 func() 中无效。  
  
5) 宏定义允许嵌套。例如：

#define PI 3.1415926

#define S PI\*y\*y /\* PI是已定义的宏名\*/

6) 习惯上宏名用大写字母表示，以便于与变量区别。但也允许用小写字母。  
  
7) 可用宏定义表示数据类型，使书写方便。例如：

#define UINT unsigned int

在程序中可用 UINT 作变量说明：

UINT a, b;

8）宏定义函数

例子：

#define SQUARE\_SUM(x,y) x\*x+y\*y

#include <stdio.h>

int main()

{

    int i = 1,j = 2,k ;

    k = SQUARE\_SUM(i+1,j);  // 这里将会替换为 i+1\*i+1+j\*j

    printf("%d",k);

    return 0;

}

很多人可能会这样算：2\*2+2\*2=8，这样其实是错误的。

原因是程序只是做了替换，也就是1+1\*1+1+2\*2=7

想达到结果为8的效果应该这样写：

#define SQUARE\_SUM(x,y) (x)\*(x)+(y)\*(y) //注意这里的括号！！

#include <stdio.h>

int main()

{

    int i = 1,j = 2,k ;

    k = SQUARE\_SUM(i+1,j);

    printf("%d",k);

    return 0;

}