c语言中的数组名和指针之间有一个重要的区别，必须要记住的是指针是变量,故p=a,p++,p--,都是有意义的操作,而数组名是常量,不是变量,因此表达式a=p,a++,a--都是非法的操作,但是&a是存在的，但是为什么编译器会对&a操作给出警告呢,但程序还是能够正常运行。Int \*p=&a这样赋值给出警告信息是因为两端的数据类型不匹配造成的,左边的p是指针类型，是一个int \*类型,而右端是一个整形的地址类型，应该将其转换为(int \*)类型.即

int \*p=(int \*)&a;

对于二维数组而言,编译系统必须能够分辨一维数组和二维数组,a[0]会被作为二维数组存储的首地址,显示的表示就是&a[0][0],“p=a;”是一维数组的表示方法,如果用在二维数组中，编译器就会给出警告信息，但是程序还是能够正常运行,在二维数组中我们可以使用“p=&a[0][0]”,但是更推荐使用”p=a[0]”的方式.

宏 与 const

在c语言中,宏(define)是一个重要的内容。无参数的宏作为常量,而带参数的宏可以提供比函数调用更加高的效率,但是自从推出了const以后，就建议使用const来代替无参数的宏的来定义常量。例如：

#define NUM 3 //常常使用如下定义来替代

Const int NUM=3;

在宏的定义中，一行只能够定义一个宏，以下的用法是错误的：

#define NUM 3, LENGTH 4

下面的这种的宏定义：

#define CH “AB” “CD” “EF”

则CH的值变为字符串ABCDEF

2.1 有参数的宏定义：

在宏的定义当中，最好使用括号将每一个参数都括起来，以免引起与优先级有关的问题,整个结果的表达式最好也用括号括起来，以防当宏用于一个更大的表达式时可能会出现同样的问题。例如：

#define ABS(x) x>0?x:-x

Printf(“%d”,ABS(2-5));

以上的宏定义应该修改为：#define ABS(x) ((x)>0?(x):-(x))

有如下的两个宏定义：

#define PSQR(x) printf(“the square of x is %d.\n”,(x)\*(x))

#define PSQR(x) printf(“the square of “#x” is %d.\n”,(x)\*(x))

我们的目标是希望在使用宏的时候能够将宏体当中的x替换为宏参数，但是第一个宏定义是不能够完成这样的目标的，因为引号当中的x会被当作是普通的文本来进行处理,我们使用第二种方法就可以实现目标.