# 程序开头结尾定式

C程序开头会有这些东西：

#include <isotream>

using namespace std;

作用：std 是一个命名空间。不同的命名空间可以有相同的类名被定义，using namespace std;就是指明下面的程序使用std,如果不用这句指明的话就要用std::string(string是std空间中定义的也可以在全局空间中定义,只要名字空间不一样即可)。否则可以默认名字空间中有std.便不用std::来修饰。这是C++新标准中有的，解决多人合作编程时名字冲突问题。如果用（旧标准）#include<iostream.h>就不需写这句话，但是如果用（新标准）#include<iostream>就必须要写。

开头结束之后，程序一般都包含这些东西：

int main() {

...//中间代码内容略

return 0;

}

这是因为定义了一个有整型返回值的主函数int main，因此要返回一个整数型。这个值一般是0，表示没有出现异常情况正常返回。比如在bat或shell脚本中，调用一个命令如何知道它执行有没有成功呢？就看它的退出码。当然也可以 return -1 或者其他值，返回非零值表示主函数执行过程中的异常情况。具体什么数字代表什么异常，可以由编程者自己定义。C中也有不返回值得主函数void main：

#include <iostream> //头文件

using namespace std; //名字空间

void main() //无返回值主函数

{

cout<<"hello world!\n"; //输出语句

}

# 函数

C语言中可以传递数组的地址给一个函数，call函数时将数组名直接写上而不加角标就是将数组地址传递给函数。但是需要注意的是，要接受数组的函数在定义时必须写清楚接受函数的类型、维度，比如下面的定义就包括了接受的数组mat的类型（char）和维度（2维），并且要注意，传递多维数组时必须指明除了第一维外的所有维度长度。

int update(char mat[][100], int mat\_len){ //向公式传递多维数组，必须指明所有维度的长度，只有第一维可以省略。

而call这个函数时就只需要将数组名写上即可（如果将角标也写上，是将数组的值复制给函数，而不是将地址传给函数）：

update(mat, n);