**CS205 C/ C++ Program Design**

**Assignment 4**

**Name**: 巫晓, **SID**: 11912803

## Part 1. Life is short, show me the code.

为了方便多个版本的迭代，我将我的代码放在了GitHub上(虽然这次只有一个version)

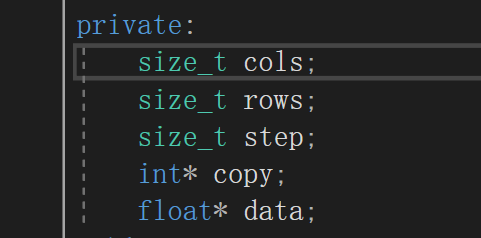
https://github.com/XiaoLing12138/cppwork.git

## Part 2. Result & Verification

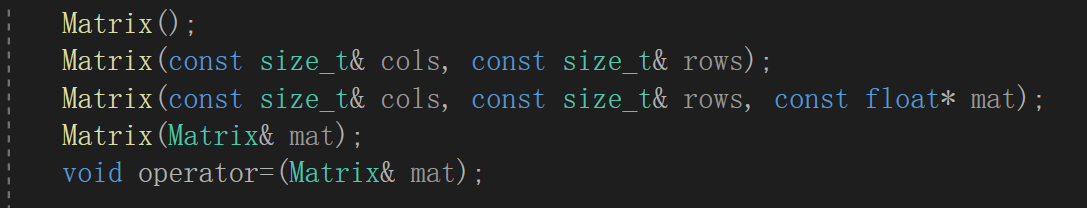
这次的project需要我们仿照opencv中的MAT简单的实现一个Matrix类。

Matrix类的声明我放在了Matrix.h中，而具体实现放在了Matrix.cpp中，test.cpp主要是用来测试。

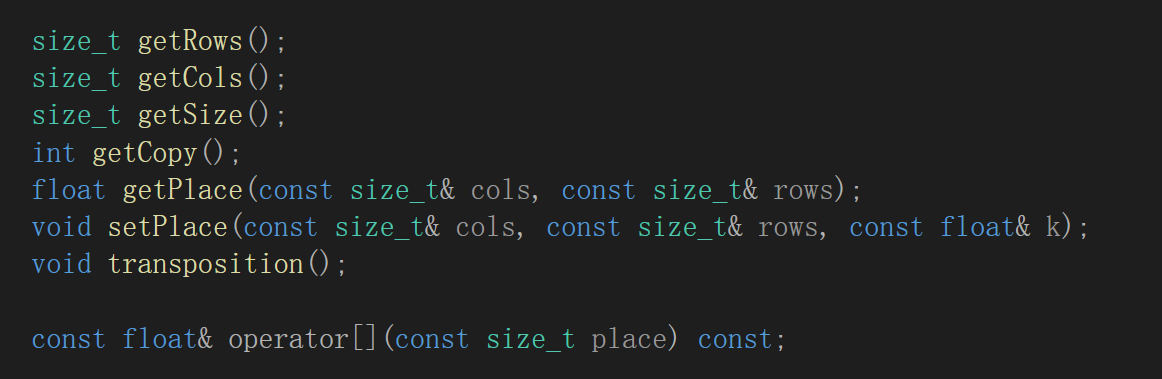
首先是定义，在Matrix类中有cols, rows, steps, \*copy, \*data。\*data指向的是矩阵起始位置（为了使用连续的内存地址存储矩阵），其中steps可以保证在大矩阵中取到小矩阵。而\*copy可以记录\*data被指向的次数，防止在执行析构函数时导致空指针错误。



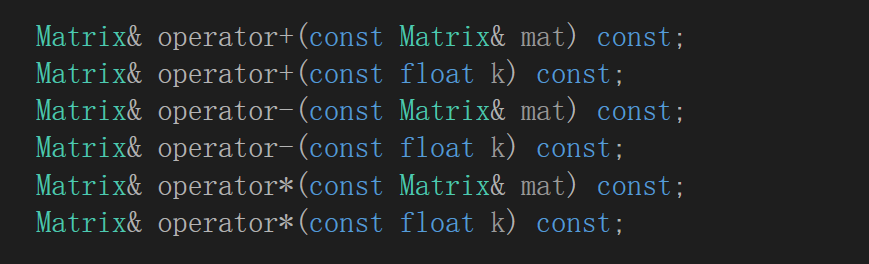
在Matrix类中，我创建了5个构造函数，可以构建空矩阵，也可以构建有初始化数据的矩阵(=)。

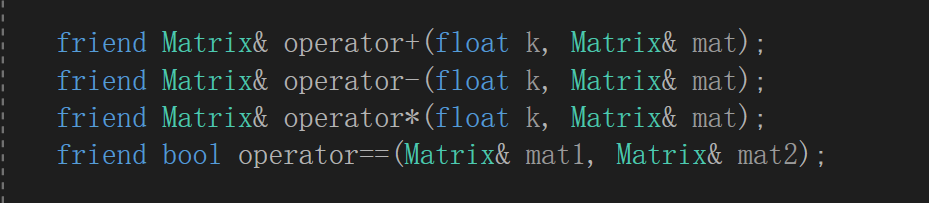


此外，还有一些比较基本的函数，用来获取矩阵信息，还有获取和更改矩阵对应位置的函数，甚至还有使自己转置的函数。

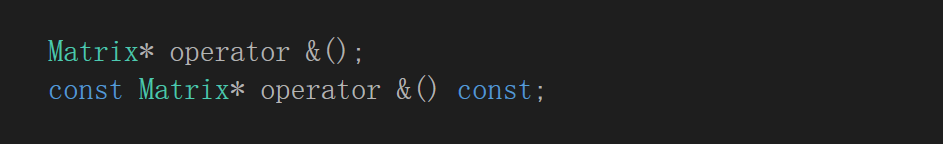


除此之外，我还实现了矩阵的运算：矩阵与数，矩阵与矩阵(向量)，还有矩阵==的判断。

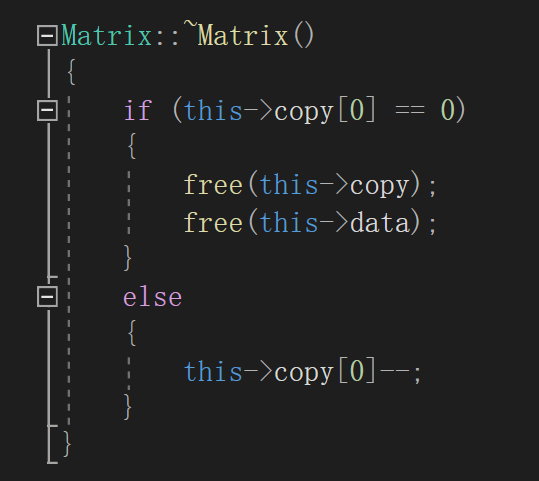




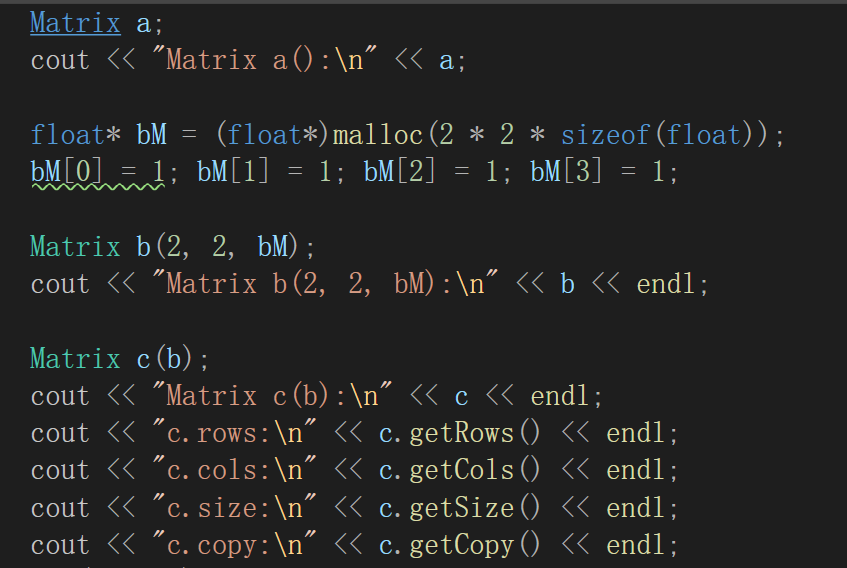
还有矩阵取地址的函数，以及适用于流输入输出的运算符重载<<。

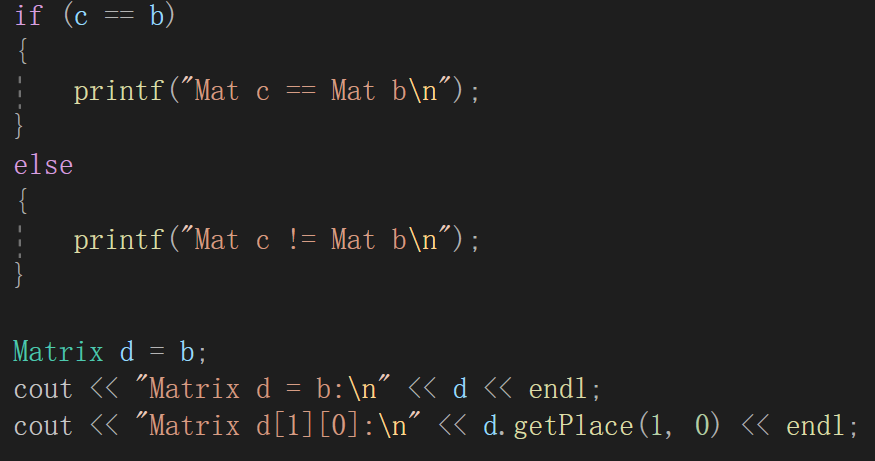


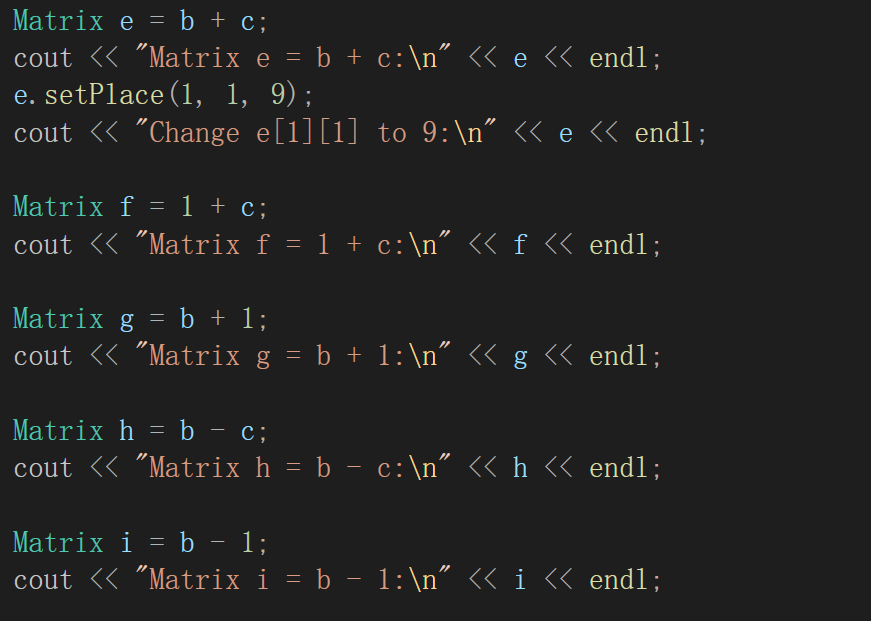
最后是析构函数，可以防止空指针异常。

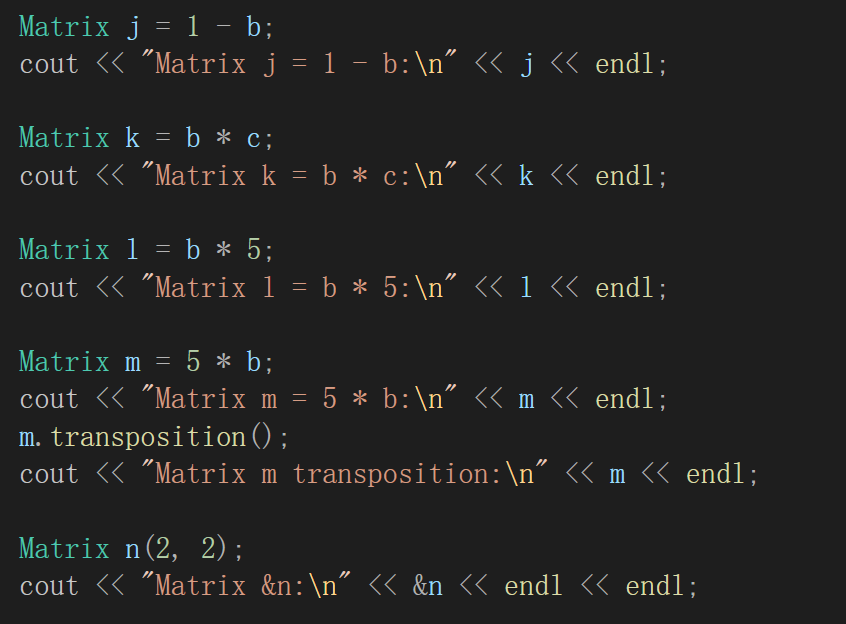


效果如下：



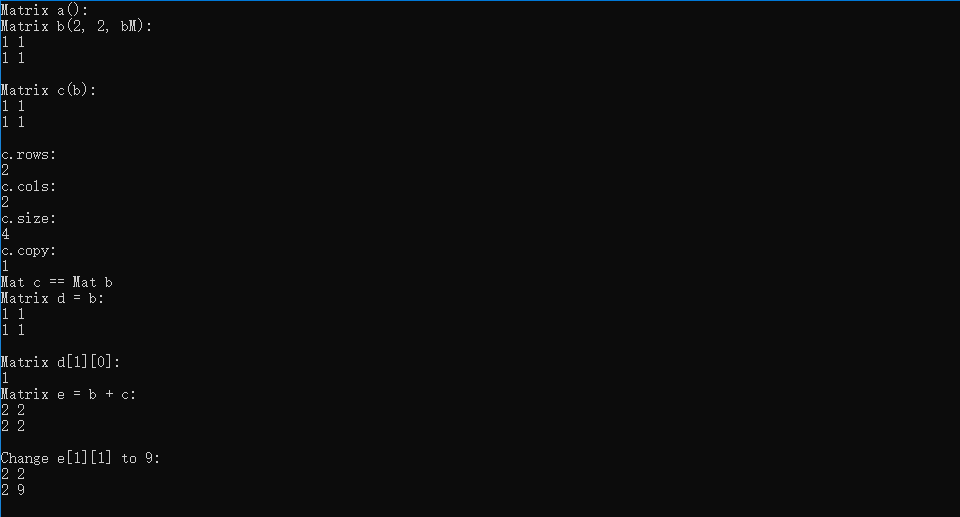








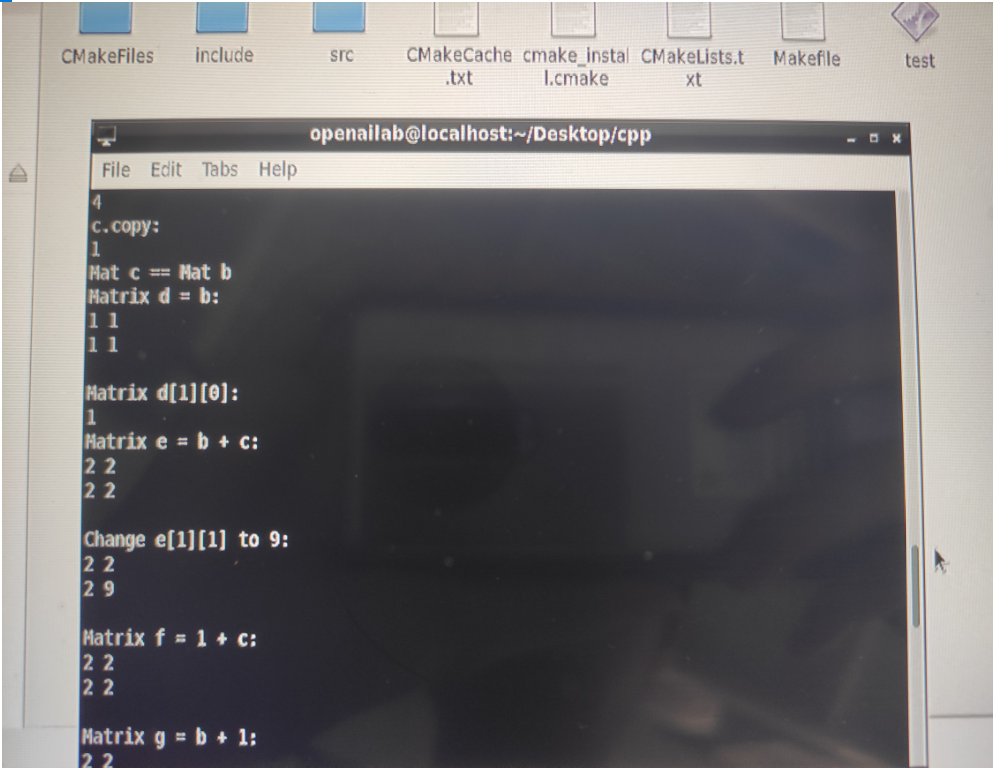
对应的结果：

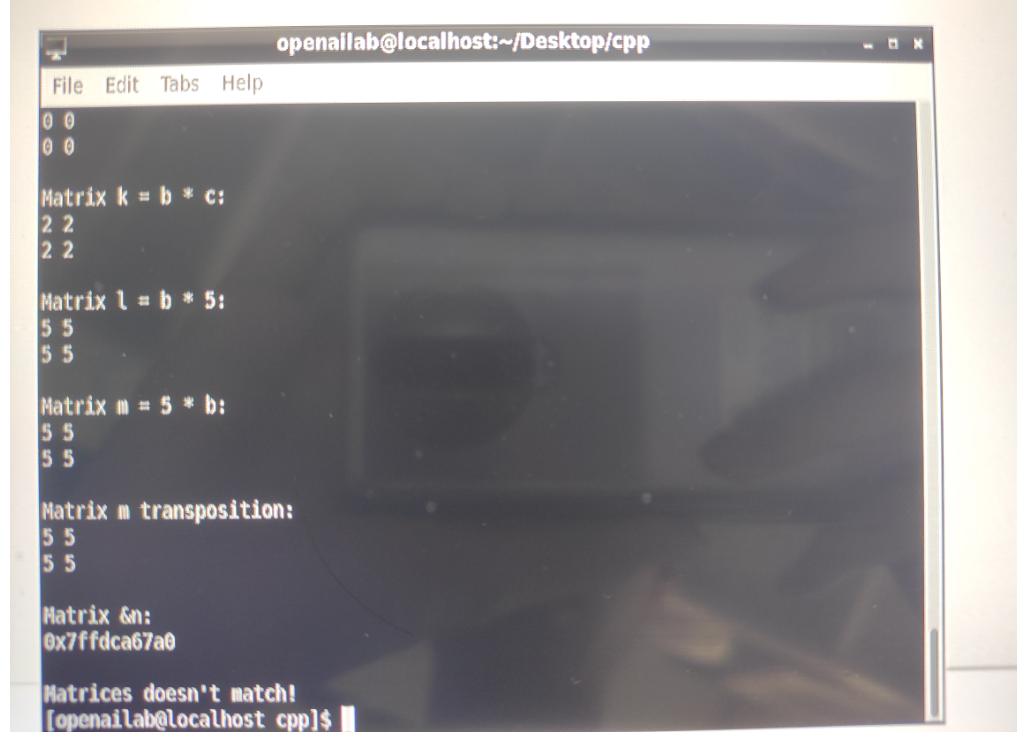






此外，我还使用了开发板进行测试，用cmake进行文件管理。





## Part 3. Difficulties & Solutions, or others

1. 难点：大矩阵的保存

解决：使用指针，用rows, cols, step记录矩阵

1. 难点：析构函数执行时导致的空指针异常

解决：用\*copy记录，\*data的复制次数，只在最后一次释放\*data

1. 难点：矩阵对应位置的替换

解决：用指针读取和替换对应位置的矩阵数据

1. 难点：矩阵运算的位置，例如A+b 与 b+A

解决：分别重载

1. 难点：矩阵输入的不合法

解决：提前判断，防止访问异常内存