使用说明：

//我们已经有了一些函数（比如生成函数，查找函数），大家可以直接用，不用再写一遍

具体功能可以打开matrix.cpp 进行翻阅

大家编写的时候最好三个文件都打开，全部选入一个project，再新建一个源文件，这样可以调用已有的函数，也可以通过main来自查写的有没有问题（仿照金组长那样发给我）

Github如果可以，就再讨论

目前已有的大部分函数都是曾写的，辛苦大家学习一遍这些函数，以方便后续各位的编程

截止至12.8 0：00

public:

string name;//名称

int No;//序号

int row, col;//行列

double \*\*data;//元素二维数组

matrix(int r = 0, int c = 0) : row(r), col(c) {} // 默认构造函数

matrix(const matrix &other); // 拷贝构造函数

matrix& operator=(const matrix &other); // 赋值运算符重载

~matrix(); // 析构函数

void matrix\_store(); // 存储函数（任意矩阵）

void matrix\_create(int row = 0, int col = 0);//生成函数（零矩阵）

void matrix\_create\_1(int order,int i, int j);//生成函数（初等矩阵1）

void matrix\_create\_2(int order,int i, int k);//生成函数（初等矩阵2）

void matrix\_create\_3(int order,int i, int j, int k);//生成函数（初等矩阵3）

void matrix\_input ();//输入函数 （二维矩阵）

void matrix\_output ();//输出函数（输出完整矩阵）

void matrix\_transpose();//输出转置矩阵并存储

double det ();

};

int matrix\_search (string na); //查找函数 （给名字返回编号）

矩阵存储方法：全局matrix类型数组list[No] ，No为编号，输入/输出的矩阵全是临时的，存储入数组后后自动销毁。

1. 函数编写规范（以查找函数为例）

// 名称：matrix\_search 查找函数（已知名字简要功能） 注释的格式就按这一套来（三行）

// 功能：查找list中的同名函数

// 参数：名字（字符串类型）

// 返回值：编号（int）

int matrix::matrix\_search(string na) { 函数命名规则：matrix\_name

bool found = false; // 用于标记是否找到矩阵 有少量关键注释即可，

for (int i = 0; i < n; i++) { 美观！（空格，换行要注意）

if (list[i].name == na) {

return i;

}

}

return -1;

}

**12.12**

新增

文沛皙：

matrix operator+(const matrix& other);// 重载加法运算符

matrix operator-(const matrix& other);// 重载减法运算符

matrix operator\*(const matrix& other);// 重载乘法运算符

matrix operator^(int k);//重载幂运算符

曾卓凡：

void matrix\_display();//输出函数（输出完整矩阵）（无需存储）

void matrix\_calculate();//计算函数（全局）

调整：

修复了matrix(const matrix &other) 的严重bug

int matrix\_search (string na); //查找函数 （给名字返回编号）（全局）

calculate函数只能加减乘，无法求幂

需求：

确认基础框架

研究分数模板

合并已有内容

讲解已有内容

检查已有内容

代码量：40 + 445 + 127

其中calculate函数占据100行

**12.17**

新增：

新函数：

徐家萌：

matrix matrix\_simplify\_1();//化简为行阶梯形矩阵

void matrix\_simplify\_2();

void matrix\_simplify\_3();

double matrix\_det();//行列式

金超：

void matrix\_create\_random(int i, int j, double a, double b);//生成函数（随机矩阵）

曾卓凡：

void transpose(); //转置函数（前端）

void det();//行列式（前端）

void output();//输出函数（前端）

void del();//删除函数

void clear();//清空函数

新功能：

cout << "delete -- to delete a matrix" << endl;

cout << "clear -- to clear all matrix" << endl;

cout << "cal -- to start calculating(+ - \* ^ T)" << endl;

cout << "det -- to calculate the determinant of your matrix " << endl;

调整：

将代码模块化：

头文件matrix.h

源文件matrix—类定义 manage—矩阵管理 calculate—矩阵计算 create—矩阵生成 main—主函数

Transpose原函数拆分为两部分（后端与前端，下同）

Det原函数拆分

Simplify\_1 返回矩阵而非void

Display 与 output函数彻底区分

Calculate函数加入转置运算，优先级，输入方式同幂运算

功能说明更新

需求：

阅读此文档，并且知晓以后修改/查找函数应该分模块来，这样不用一下子传一堆文件

分数需求（待定）（这个实在不行就曾卓凡来）

一名专职找bug的组员（待定）（文/徐？ 确定之后可以和曾卓凡详谈）

Modify模块（金超）（可加入增广矩阵功能？）

Create模块（随机对角矩阵，上三角矩阵，单位矩阵）（和后两个比算简单的，有意愿者可以练手）

逆矩阵（待定）

解线性方程（待定）（这俩不着急）

Main完善（这个曾卓凡来吧）