通讯录设计实验报告

实验日期：**2021.11.6** 班级：计科**201**

学号**1**：**20402070113**姓名**1**：瞿子椋

学号**2**：**20402080240**姓名**2**：王夏禹

学号**3**：**20401010121**姓名**3**：徐志伟

# 实验目的

了解如何计算矩阵的加乘一句以及基本算法并能加以运用。

# 二、实验环境

一台使用**visualc++6.0/CFree**的计算机

# 三、实验内容

求对称矩阵之和、乘积

已知A和B为两个n×n阶的对称矩阵，编写一个程序实现：

1. 将其下三角元素存储在一维数组a和b中，并输出。

1 1 2 4

1 2 3 5

2 3 4 6

4 5 6 7

1 1 1 1

1 1 1 1

1 1 1 1

1 1 1 1

提示：程序中可以用int A[4][4]和B[4][4]二维数组表示原始矩阵A和B。

1. 设C＝A＋B，以矩阵方式输出C。
2. 设D＝A×B，以矩阵方式输出D。

（注意：将该题翻译成英文）

# 四、重要的数据结构

**int** getFileRows(**const** **char** \*fileName)

{

ifstream fileStream;

string tmp;

**int** count = 0;// 行数计数器

fileStream.open(fileName,ios::in);//ios::in 表示以只读的方式读取文件

**if**(fileStream.fail())//文件打开失败:返回0

{

**return** 0;

}

**else**//文件存在

{

**while**(getline(fileStream,tmp,'\n'))//读取一行

{

**if** (tmp.size() > 0 )

count++;

}

fileStream.close();

**return** count ;

}

}

//sum

**if**(Row1==Row2)

{

**int** Ansewer1[Row2][Row2];

**for** (**int** i = 0; i < Row1; i++)

{

**for** (**int** j = 0; j < Row1; j++)

{

Ansewer1[i][j] =Matrix1[i][j] + Matrix2[i][j];

}

}

cout<<"两矩阵相加结果："<<endl;

**for** (**int** i = 0; i < Row1; i++)

{

**for** (**int** j = 0; j < Row1; j++)

{

cout << Ansewer1[i][j]<<" ";

}

cout << endl;

}

}

**else**

{

cout<<"两矩阵不满足矩阵加法运算条件"<<endl;

}

//mutiply

**if**(Row1==Row2)

{

**int** Ansewer2[Row2][Row2];

**for** (**int** i = 0; i < Row1; i++)

{

**for** (**int** j = 0; j < Row1; j++)

{

Ansewer2[i][j] = 0;

**for** (**int** k = 0; k < Row1; k++)

{

Ansewer2[i][j] += Matrix1[i][k] \* Matrix2[k][j];

}

}

}

cout<<"两矩阵相乘结果："<<endl;

**for** (**int** i = 0; i < Row1; i++)

{

**for** (**int** j = 0; j < Row1; j++)

{

cout << Ansewer2[i][j]<<" ";

}

cout << endl;

}

}

**else**

{

cout<<"两矩阵不满足矩阵乘法运算条件"<<endl;

}

}

# 实施分析

先将数据存储在**nums.txt**文件中然后在计算**A**矩阵和**B**矩阵的和并以矩阵**C**输出

后在计算**A**矩阵和**B**矩阵的乘积并以矩阵**D**输出。

# 六.调试中遇到的问题及解决方法

一开始无法从文件中读出矩阵的行数，后来通过以下算法得出：

**int** getFileRows(**const** **char** \*fileName)

{

ifstream fileStream;

string tmp;

**int** count = 0;// 行数计数器

fileStream.open(fileName,ios::in);//ios::in 表示以只读的方式读取文件

**if**(fileStream.fail())//文件打开失败:返回0

{

**return** 0;

}

**else**//文件存在

{

**while**(getline(fileStream,tmp,'\n'))//读取一行

{

**if** (tmp.size() > 0 )

count++;

}

fileStream.close();

**return** count ;

}

}

然后为了能将文件中矩阵的数据存储在数组中进行计算，只能取消全局函数的形式，将函数中的算法在main（）里实现。

# 七、总结

在此次实验中我们学习到了如何用计算机语言计算对称矩阵的和和乘积，大大锻炼了我们的编程能力，让我们今后能更加游刃有余的编写我们所需要的代码。

# 八、工作划分

|  |  |  |
| --- | --- | --- |
|  | 分组 |  |
| 成员名称 | 完成的工作 | 完成情况 |
| 瞿子椋 | 计算对称矩阵的和 | 已完成 |
| 王夏禹 | 计算对称矩阵的乘积 | 已完成 |
| 徐志伟 | 完成实验报告 | 已完成 |