CS205 C/C++ Programming - Project 5

Name: 钟元吉(Zhong Yuanji)

SID: 12012613

Part 1 - Analysis

> Design a class for matrices, and the class should contain the data of a matrix and related information such the number of rows, the number of columns, the number of channels, etc. The Matrix can support different data types, use soft or hard copy to manage memory and support operators =, ==, !=, <, >, <=, >=, +, -, *, /, ^.

本次Project要求我们仅通过C++语言建立对不同数据格式的、支持行列与通道数的矩阵类，实现矩阵软拷贝和硬拷贝的内存管理，实现运算符重载和矩阵类型转换。

Part 2 - Code

文件说明

文件名	内容解释
Matrix.hpp	矩阵类及其函数头、友元函数头
Matrix.cpp	矩阵类函数定义、友元函数定义
Test0.cpp	测试文件 - 基本要求的实现
Test1.cpp	测试文件 - 对错误类型的检查
Test2.cpp	测试文件 - 其他特殊功能

部分代码展示

由于代码较长，这里只展示**通过字符串构造矩阵**和**计算矩阵的逆**的部分。

```
// Create a matrix from string
_TP _MAT::Matrix(const char *strOrg)
{
    // Check if string is empty
    if (!strOrg)
    {
        __PRINT_ERROR("The input string is NULL when initializing a matrix.");
        return;
    }
    // Copy the string
    size_t orgLen = strlen(strOrg);
    char str[orgLen];
    strcpy(str, strOrg);
    // Replace the blanks into ','
    for (size_t i = 0, j = 0; j <= orgLen; ++j)
        if ((str[j] == ',' || str[j] == ';') && str[i - 1] == ',')
            str[i - 1] = str[j];
        else if (str[j] != ' ')
            str[i++] = str[j];
        else if (j != 0 && ((str[j - 1] >= '0' && str[j - 1] <= '9') || str[j - 1] == '.'))
            str[i++] = ',';
    size_t len = strlen(str);
    // Check format error
    if (len < 3 || str[0] != '[' || str[1] == ',' || str[1] == ';' || str[1] == ']' || str[len - 1] != ']')
    {
        __PRINT_ERROR("The input string must be with the form [ ,;] when initializing a matrix.");
        return;
    }
    while (str[len - 1] == ',' || str[len - 1] == ';' || str[len - 1] == ']')
        str[--len] = '\0';
    // Count rows and cols
    size_t countD = 0, countF = 0;
    for (size_t i = 1; i < len; ++i)
        if (str[i] == ';')
            ++countF;
        else if (str[i] == ',' && !countF)
            ++countD;
    // Read from string
    size_t At = 1;
    row = countF + 1, col = countD + 1;
    lines = col, channel = 1;
    ref = new int(1);
    data = new _T[row * col], at = data;
    for (size_t i = 0; i <= countF; ++i)
        for (size_t j = 0; j <= countD; ++j, ++At)
        {
            double temp;
            sscanf(&str[At], "%lf", &temp);
            at[i * lines + j] = (_T)temp;
            for (; At < len; ++At)
            {
                if (str[At] == ',')
                {
                    if (str[At + 1] == ';')
                        ++At;
                    break;
                }
                else if (str[At] == ';')
                {
                    if (j != countD)
                        goto RETURN_Error;
                    break;
                }
                else if ((str[At] < '0' || str[At] > '9') && str[At] != '.' && str[At] != '-' && str[At] != 'e' && str[At] != 'E')
                    goto RETURN_Error;
            }
        }
    if (At < len)
        goto RETURN_Error;
    return;
RETURN_Error:
    __PRINT_ERROR("The input string is not valid when initializing a matrix.");
    clear();
}
// ========================================================================
// Compute the inverse
_TP Matrix<long double> _MAT::inv(size_t chaAt) const
{
    // Check if it is empty
    if (!row || !col || !channel || !ref)
    {
        __PRINT_ERROR("Empty Matrix has no inverse.");
        return Matrix<long double>();
    }
    if (row != col)
    {
        __PRINT_ERROR("Only square Matrix has inverse.");
        return Matrix<long double>();
    }
    // If the size of matrix is 1x1
    if (row == 1)
    {
        if (!at[0])
        {
            __PRINT_ERROR("The Matrix is not invertible.");
            return Matrix<long double>();
        }
        long double ret = 1.L / at[chaAt * row * lines];
        return Matrix<long double>(1, 1, new long double[1]{ret}, 1);
    }
    // If the size of matrix is not 1x1
    long double *data_ = new long double[row * col], matDet = det(chaAt);
    if (!matDet)
    {
        __PRINT_ERROR("The Matrix is not invertible.");
        return Matrix<long double>();
    }
#pragma omp parallel for
    for (size_t i = 0; i < row; ++i)
        for (size_t j = 0; j < col; ++j)
        {
            Matrix tmp = cofactorMatrix(j, i);
            data_[i * col + j] = ((i + j) % 2 ? -tmp.det() : tmp.det()) / matDet;
        }
    return Matrix<long double>(row, col, data_, 1);
}
```

Part 3 - Result & Verification

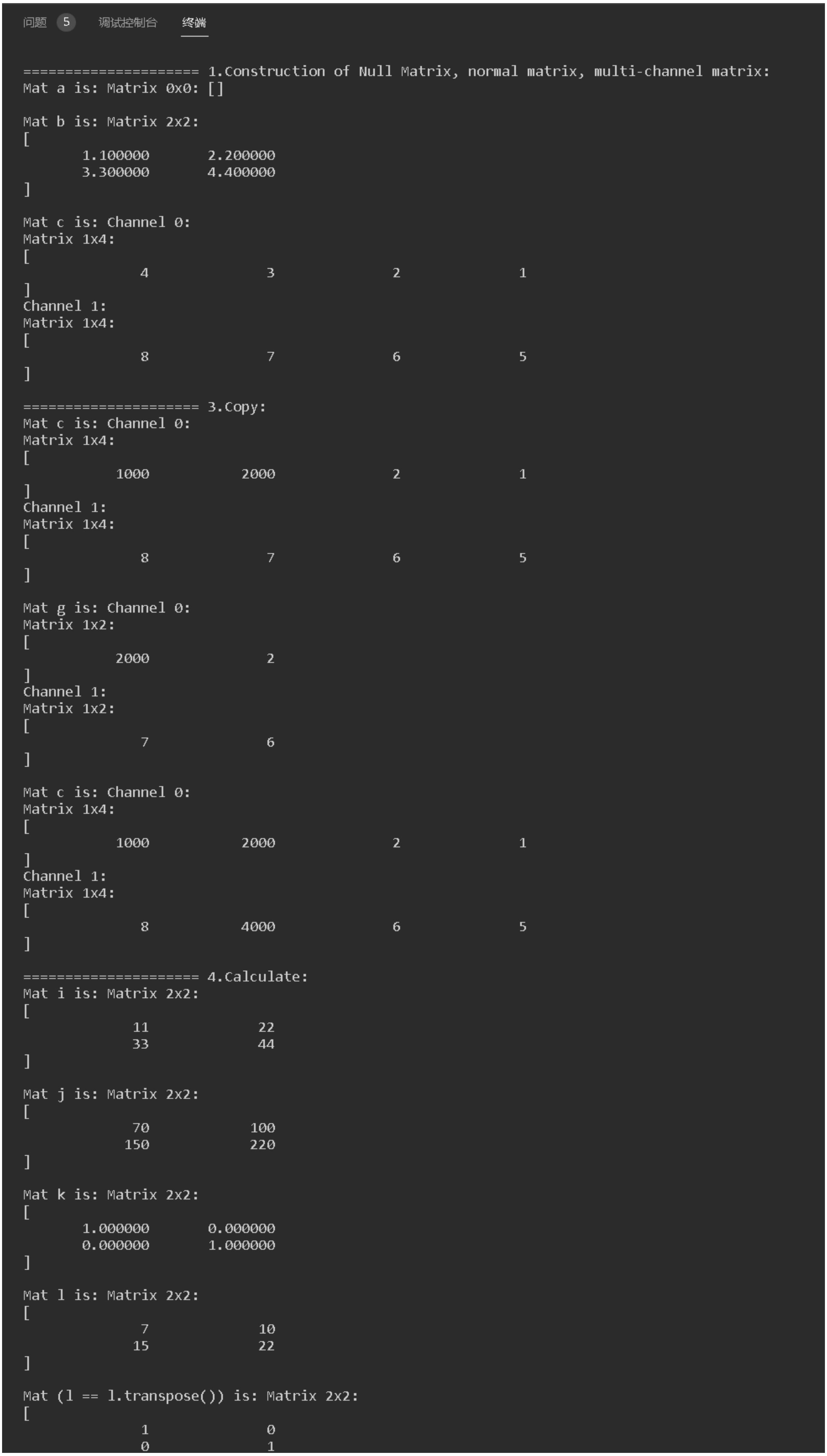
Test case #1: 基本要求的实现

直接编译并运行`Test0.cpp`，或使用`makefile`编译并运行：

```
make
```

```
g++ ./src/Test0.cpp -o Test0 -w & ./Test0
```

1. 构造int空矩阵a、double单通道矩阵b、size_t双通道矩阵c；
2. 通过改变赋值后矩阵d=c(软拷贝)，构造复制矩阵e(c)(软拷贝)，硬拷贝f=c.copy()、子矩阵g=c.sub(...)(软拷贝)、硬拷贝子矩阵h=c.subCopy()的元素，查看是否同时改变了c来检查软硬拷贝；
3. 展示计算功能，其中`A/B = A * B.inv()`, `A^2 = A * A`, `B^-1 = B.inv()`, `C^0 = Id`, 完全不对称矩阵有`A==A.transpose() -> Id`

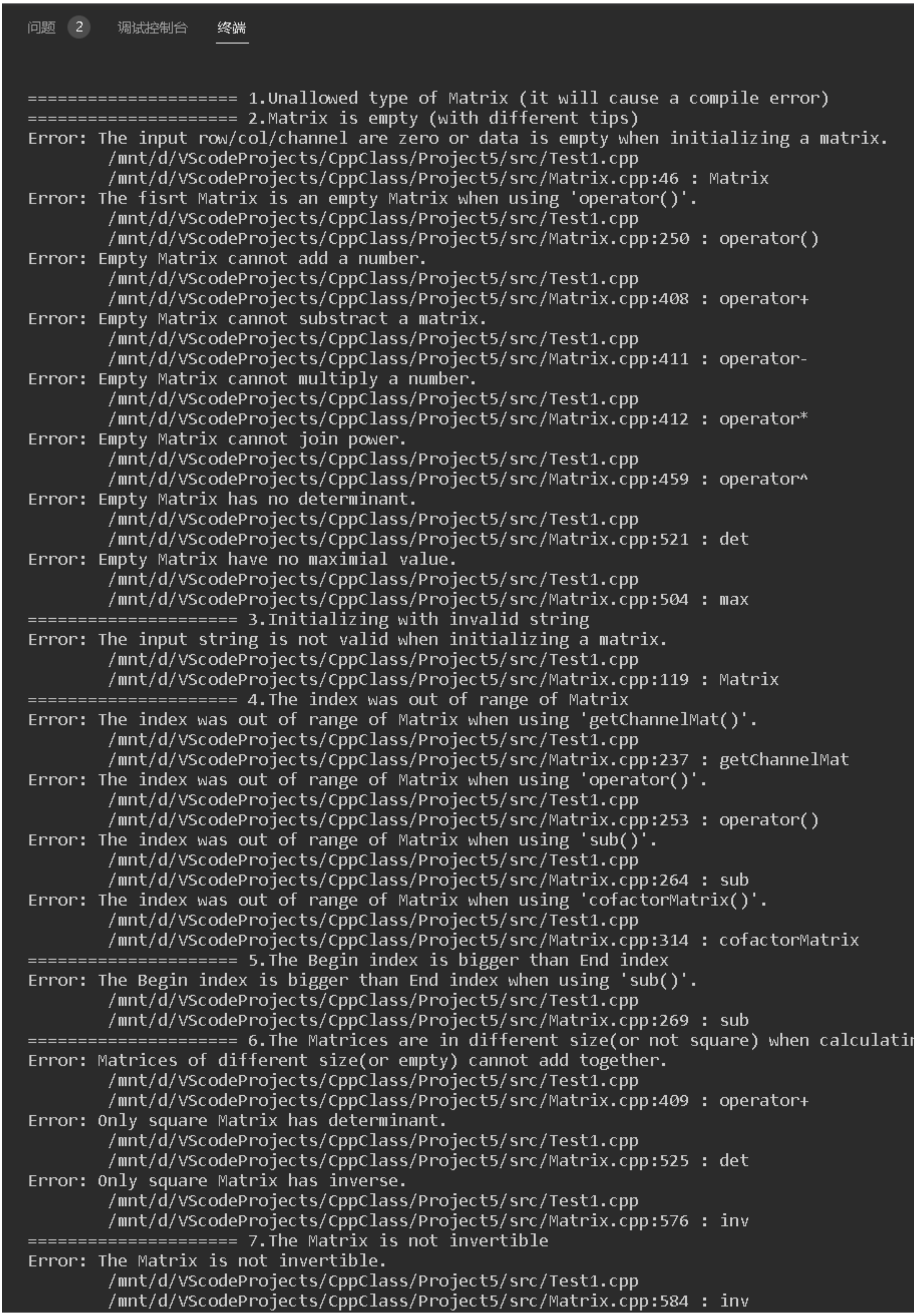


```
==================== 1.Construction of Null Matrix, normal matrix, multi-channel matrix:
Mat a is: Matrix 0x0: []

Mat b is: Matrix 2x2:
[
       1.100000       2.200000
       3.300000       4.400000
]

Mat c is: Channel 0:
Matrix 1x4:
[
              4              3              2              1
]
Channel 1:
Matrix 1x4:
[
              8              7              6              5
]

==================== 3.Copy:
Mat c is: Channel 0:
Matrix 1x4:
[
           1000           2000              2              1
]
Channel 1:
Matrix 1x4:
[
              8              7              6              5
]

Mat g is: Channel 0:
Matrix 1x2:
[
           2000              2
]
Channel 1:
Matrix 1x2:
[
              7              6
]

Mat c is: Channel 0:
Matrix 1x4:
[
           1000           2000              2              1
]
Channel 1:
Matrix 1x4:
[
              8           4000              6              5
]

==================== 4.Calculate:
Mat i is: Matrix 2x2:
[
             11             22
             33             44
]

Mat j is: Matrix 2x2:
[
             70            100
            150            220
]

Mat k is: Matrix 2x2:
[
       1.000000       0.000000
       0.000000       1.000000
]

Mat l is: Matrix 2x2:
[
              7             10
             15             22
]

Mat (l == l.transpose()) is: Matrix 2x2:
[
              1              0
              0              1
]
```

Test case #2: 对错误类型的检查

直接编译并运行`Test1.cpp`：

```
g++ ./src/Test1.cpp -o Test1 -w & ./Test1
```



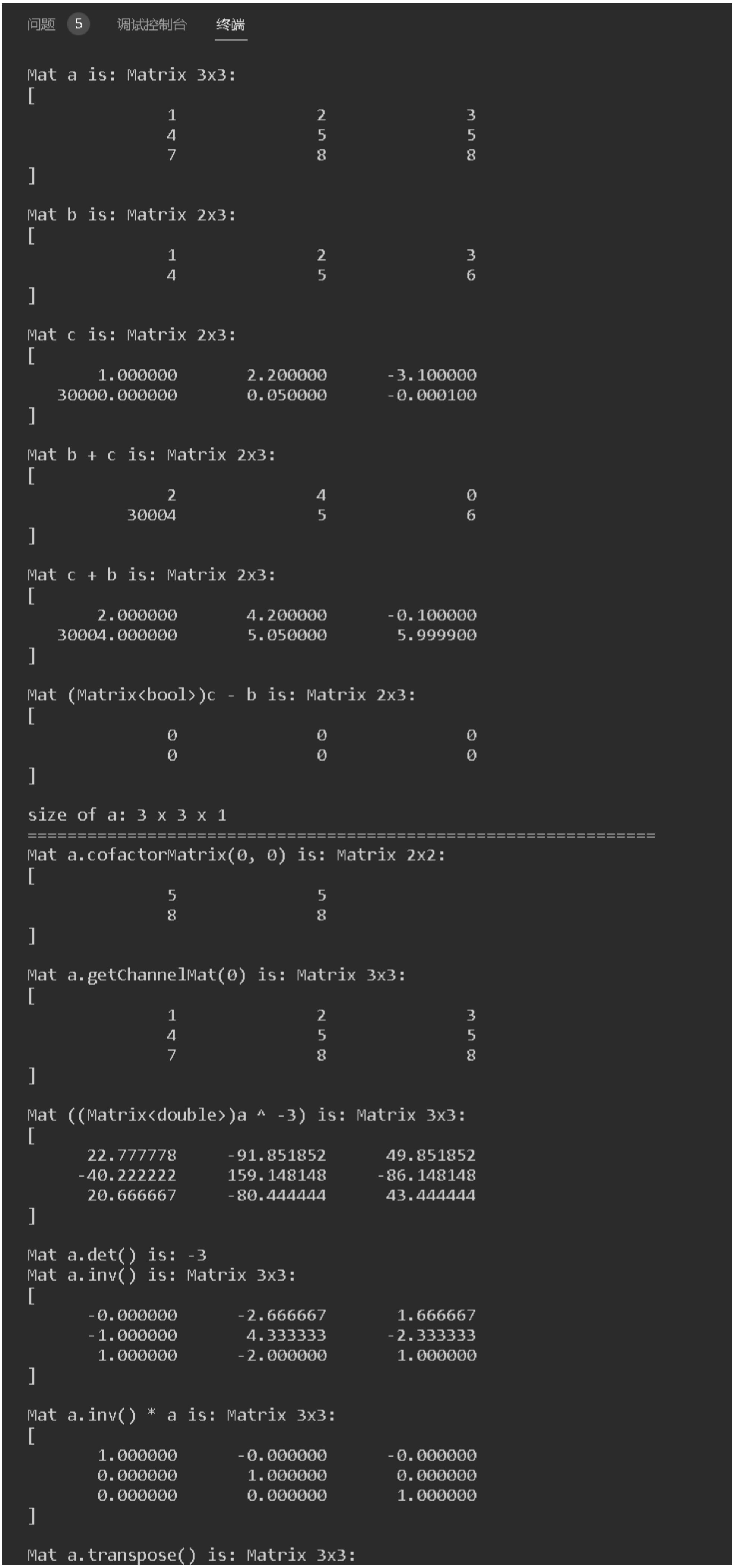
```
===================== 1.Unallowed type of Matrix (it will cause a compile error)
===================== 2.Matrix is empty (with different tips)
Error: The input row/col/channel are zero or data is empty when initializing a matrix.
        /mnt/d/VScodeProjects/CppClass/Project5/src/Test1.cpp
        /mnt/d/VScodeProjects/CppClass/Project5/src/Matrix.cpp:46 : Matrix
Error: The fisrt Matrix is an empty Matrix when using 'operator()'.
        /mnt/d/VScodeProjects/CppClass/Project5/src/Test1.cpp
        /mnt/d/VScodeProjects/CppClass/Project5/src/Matrix.cpp:250 : operator()
Error: Empty Matrix cannot add a number.
        /mnt/d/VScodeProjects/CppClass/Project5/src/Test1.cpp
        /mnt/d/VScodeProjects/CppClass/Project5/src/Matrix.cpp:408 : operator+
Error: Empty Matrix cannot substract a matrix.
        /mnt/d/VScodeProjects/CppClass/Project5/src/Test1.cpp
        /mnt/d/VScodeProjects/CppClass/Project5/src/Matrix.cpp:411 : operator-
Error: Empty Matrix cannot multiply a number.
        /mnt/d/VScodeProjects/CppClass/Project5/src/Test1.cpp
        /mnt/d/VScodeProjects/CppClass/Project5/src/Matrix.cpp:412 : operator*
Error: Empty Matrix cannot join power.
        /mnt/d/VScodeProjects/CppClass/Project5/src/Test1.cpp
        /mnt/d/VScodeProjects/CppClass/Project5/src/Matrix.cpp:459 : operator^
Error: Empty Matrix has no determinant.
        /mnt/d/VScodeProjects/CppClass/Project5/src/Test1.cpp
        /mnt/d/VScodeProjects/CppClass/Project5/src/Matrix.cpp:521 : det
Error: Empty Matrix have no maximial value.
        /mnt/d/VScodeProjects/CppClass/Project5/src/Test1.cpp
        /mnt/d/VScodeProjects/CppClass/Project5/src/Matrix.cpp:504 : max
===================== 3.Initializing with invalid string
Error: The input string is not valid when initializing a matrix.
        /mnt/d/VScodeProjects/CppClass/Project5/src/Test1.cpp
        /mnt/d/VScodeProjects/CppClass/Project5/src/Matrix.cpp:119 : Matrix
===================== 4.The index was out of range of Matrix
Error: The index was out of range of Matrix when using 'getChannelMat()'.
        /mnt/d/VScodeProjects/CppClass/Project5/src/Test1.cpp
        /mnt/d/VScodeProjects/CppClass/Project5/src/Matrix.cpp:237 : getChannelMat
Error: The index was out of range of Matrix when using 'operator()'.
        /mnt/d/VScodeProjects/CppClass/Project5/src/Test1.cpp
        /mnt/d/VScodeProjects/CppClass/Project5/src/Matrix.cpp:253 : operator()
Error: The index was out of range of Matrix when using 'sub()'.
        /mnt/d/VScodeProjects/CppClass/Project5/src/Test1.cpp
        /mnt/d/VScodeProjects/CppClass/Project5/src/Matrix.cpp:264 : sub
Error: The index was out of range of Matrix when using 'cofactorMatrix()'.
        /mnt/d/VScodeProjects/CppClass/Project5/src/Test1.cpp
        /mnt/d/VScodeProjects/CppClass/Project5/src/Matrix.cpp:314 : cofactorMatrix
===================== 5.The Begin index is bigger than End index
Error: The Begin index is bigger than End index when using 'sub()'.
        /mnt/d/VScodeProjects/CppClass/Project5/src/Test1.cpp
        /mnt/d/VScodeProjects/CppClass/Project5/src/Matrix.cpp:269 : sub
===================== 6.The Matrices are in different size(or not square) when calculati
Error: Matrices of different size(or empty) cannot add together.
        /mnt/d/VScodeProjects/CppClass/Project5/src/Test1.cpp
        /mnt/d/VScodeProjects/CppClass/Project5/src/Matrix.cpp:409 : operator+
Error: Only square Matrix has determinant.
        /mnt/d/VScodeProjects/CppClass/Project5/src/Test1.cpp
        /mnt/d/VScodeProjects/CppClass/Project5/src/Matrix.cpp:525 : det
Error: Only square Matrix has inverse.
        /mnt/d/VScodeProjects/CppClass/Project5/src/Test1.cpp
        /mnt/d/VScodeProjects/CppClass/Project5/src/Matrix.cpp:576 : inv
===================== 7.The Matrix is not invertible
Error: The Matrix is not invertible.
        /mnt/d/VScodeProjects/CppClass/Project5/src/Test1.cpp
        /mnt/d/VScodeProjects/CppClass/Project5/src/Matrix.cpp:584 : inv
```

Test case #3: 其他特殊功能

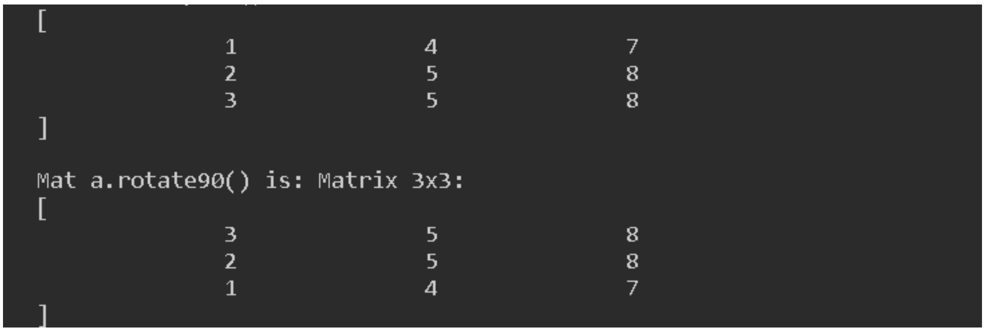
直接编译并运行`Test2.cpp`：

```
g++ ./src/Test2.cpp -o Test2 -w & ./Test2
```

1. 通过字符串构造单通道矩阵，只能识别空格(','列分割，';'行分割，空格在无逗号时代替逗号)；
2. 自动转换矩阵的数据类型，计算时转换其后矩阵为与第一个出现的矩阵相同的类型；
3. 通过圆括号索引`(row,col,channel=0)`(可写), `getRow()`, `getCol()`, `getChannel()`, `getChannelMat()`(可写)返回私有变量的访问；
4. 特殊函数：余子式矩阵、返回指定通道矩阵、矩阵整数幂运算(快速幂法)、行列式、逆矩阵、转置、逆时针旋转90度。



```
Mat a is: Matrix 3x3:
[
              1              2              3
              4              5              5
              7              8              8
]

Mat b is: Matrix 2x3:
[
              1              2              3
              4              5              6
]

Mat c is: Matrix 2x3:
[
       1.000000       2.200000      -3.100000
   30000.000000       0.050000      -0.000100
]

Mat b + c is: Matrix 2x3:
[
              2              4              0
          30004              5              6
]

Mat c + b is: Matrix 2x3:
[
       2.000000       4.200000      -0.100000
   30004.000000       5.050000       5.999900
]

Mat (Matrix<bool>)c - b is: Matrix 2x3:
[
              0              0              0
              0              0              0
]

size of a: 3 x 3 x 1
================================================================
Mat a.cofactorMatrix(0, 0) is: Matrix 2x2:
[
              5              5
              8              8
]

Mat a.getChannelMat(0) is: Matrix 3x3:
[
              1              2              3
              4              5              5
              7              8              8
]

Mat ((Matrix<double>)a ^ -3) is: Matrix 3x3:
[
      22.777778     -91.851852      49.851852
     -40.222222     159.148148     -86.148148
      20.666667     -80.444444      43.444444
]

Mat a.det() is: -3
Mat a.inv() is: Matrix 3x3:
[
      -0.000000      -2.666667       1.666667
      -1.000000       4.333333      -2.333333
       1.000000      -2.000000       1.000000
]

Mat a.inv() * a is: Matrix 3x3:
[
       1.000000      -0.000000      -0.000000
       0.000000       1.000000       0.000000
       0.000000       0.000000       1.000000
]

Mat a.transpose() is: Matrix 3x3:
```



```
[
              1              4              7
              2              5              8
              3              5              8
]

Mat a.rotate90() is: Matrix 3x3:
[
              3              5              8
              2              5              8
              1              4              7
]
```

Part 4 - Difficulties & Solutions

问题：

Part 5 - Summary

以上是本次Report的所有内容，感谢您的阅读！