Project 1

<Matrix>

<Version 1.0>

Name: Haolan Ye (Benjamin)

Class: CSC-17C 48942

Date: 11/16/2015

**Contents**

1. **Introduction……………………………………...2**
2. **Summary……………………………………........2**
3. **Problems during coding…………………………3**
4. **Pseudo Code…………………………………...…4**
5. **Screen shot…………..…………..…………..........5**
6. **System Libraries……………………………...…10**
7. **Concept Covered…..…….……..……..………....11**
8. **Flowchart...…………..…………………………..13**
9. **Introduction(**[**https://en.wikipedia.org/wiki/Matrix\_(mathematics)**](https://en.wikipedia.org/wiki/Matrix_(mathematics)) **):**

In mathematics, a matrix (plural matrices) is a rectangular array—of numbers, symbols, or expressions, arranged in rows and columns—that is interpreted and manipulated in certain prescribed ways. One such way is to state the dimensions of the matrix.

In this program, the main program is the calculator including function add/ subtract /multiply two matrices, and the transpose, determinant, and inverse of matrix. In addition, there is an easy matrix game for users to play.

1. **Summary:**

|  |  |
| --- | --- |
| Total Line of Code | 1000+ |
| Comment Line | - |
| Variable | - |
| Function | - |

This game contains most concepts that we have learned in the class. I have my own Linked List template class which is used in Record Class. There are also file stream and quick sort in Record Class. I use the STL library to get a statistic of all the records. In the Matrix Class, I use recursion to solve the determinant problems. I use the 2-dimention array to store the matrix.

1. **Problems during coding**
2. **Get the determinant**

At the beginning, it’s hard for me to get the determinant. Therefore, I divided into some parts. First, I have a function to get the cofactor of a position. Then, I calculate the determinant with cofactors.

1. **Sort with my own Linked List**

In the Record Class, I use my own linked list to store the record inputted from the file. In the quick sort function parameter, I didn’t use reference variables to my linked list. Thus, I tried so many times, the result of sort is the original record. Afterward, I realized that I didn’t use the reference variables, and I fixed it.

When I did the project two weeks ago, I wanted to use my own linked list to store the matrix. After I append a matrix class into the list, the content of the class is changed. Two days later, I realized that the copy constructor and “=” (overload operator) have to be utilized in the append function of my list class. Therefore, I fixed that problems after I added copy constructor and “=” (overload operator) in my Matrix Class.

1. **Formatted Output the result**

Since the element of matrix might have more one digit, I need to formatted output the matrix. In addition, In the Record Class, the names of players have different lengths, so I should also formatted output the name and the record relative to the name.

1. **Pseudo Code**

Set seed for random number

Display main menu

If choose Matrix Calculator

Get into calculator function

If choose Game

Get into Game Function

If choose Exit

Exit the program

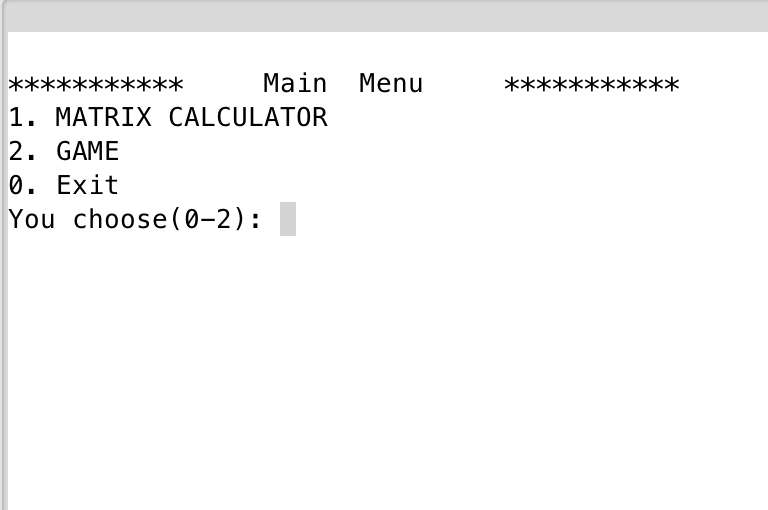
Matrix Calculator

1. Ask user what he need to calculate (add, subtract, multiply, inverse, determinant, transpose)
2. Get the matrix/matrices from the user
3. Calculate and output

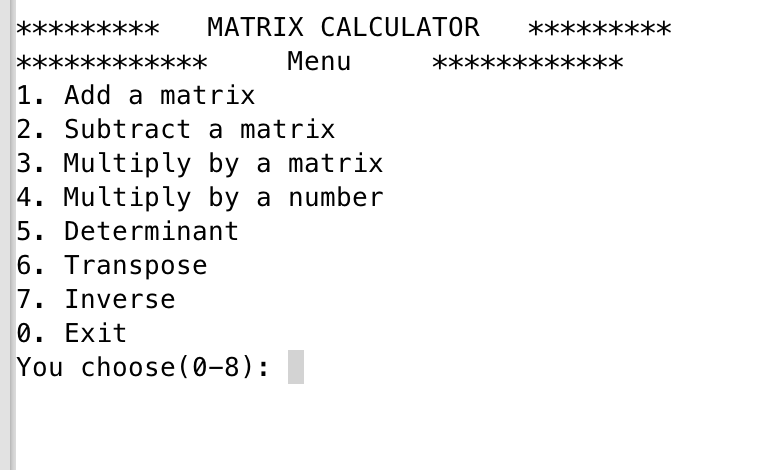
Game

1. Randomly create a 2\*2 matrix, and 5 times randomly choose a question from determinant, multiply by a number, transpose, a cofactor of a position, and multiply by a matrix.
2. Ask player for the answer
3. Compare the answer inputted and the real answer
4. Call the Record Class
5. Input the record
6. Quick sort records
7. Display records

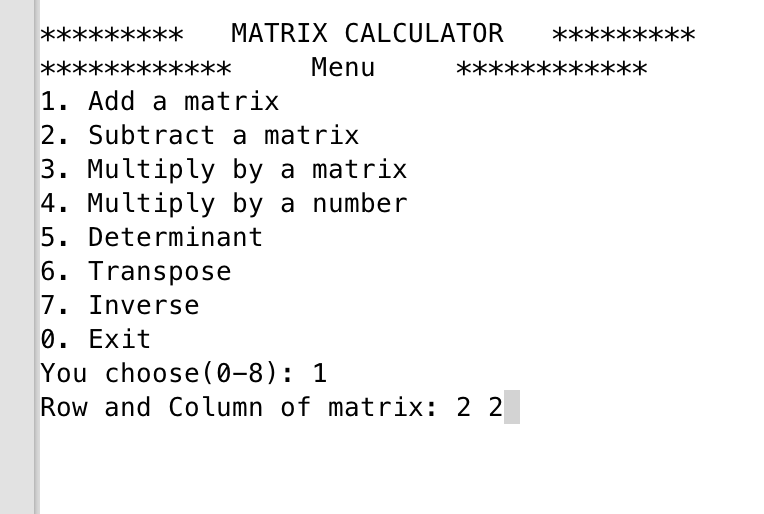
1. **Screen Shot**
2. Main menu



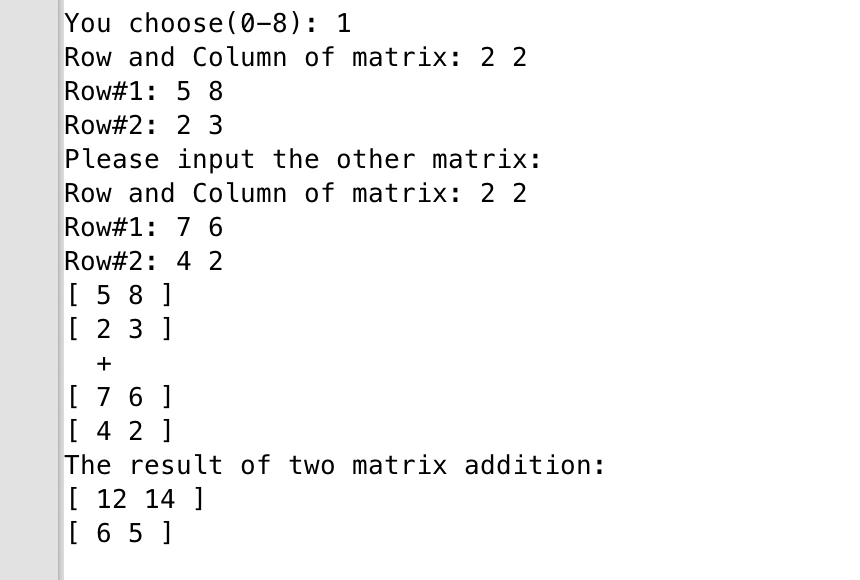
1. After choosing 1, get into matrix calculator



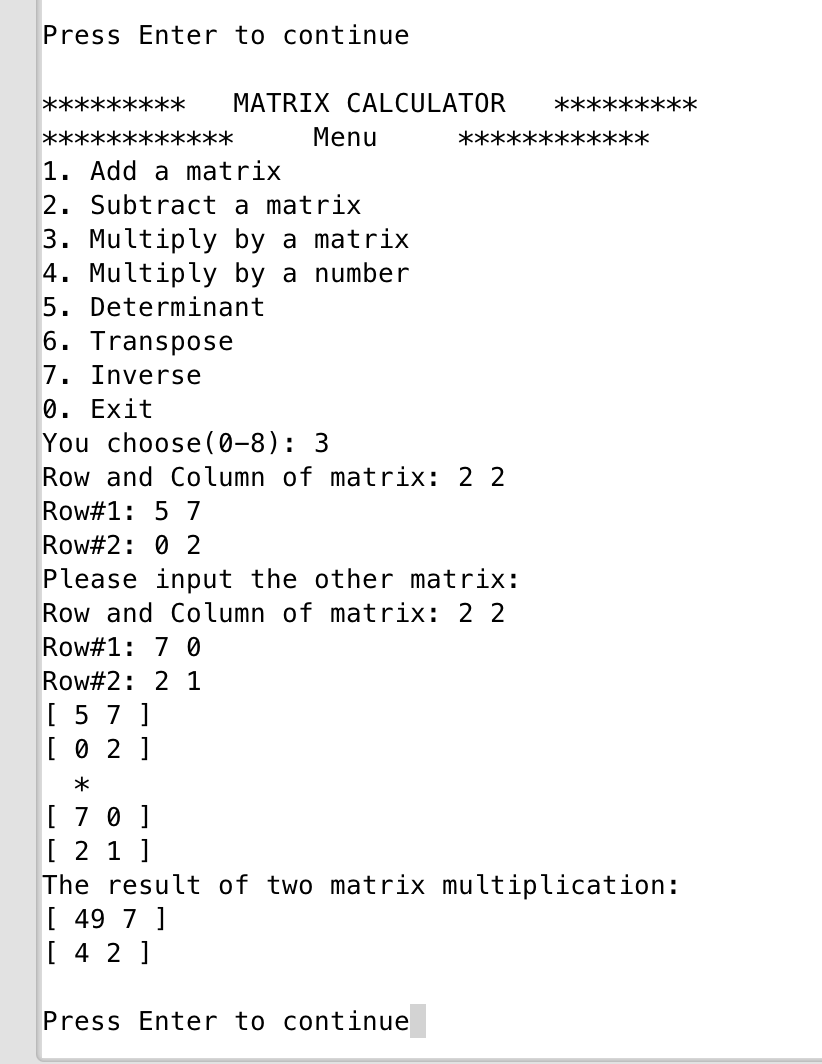
1. Choose 1 for add a matrix. Input the row and column of matrix



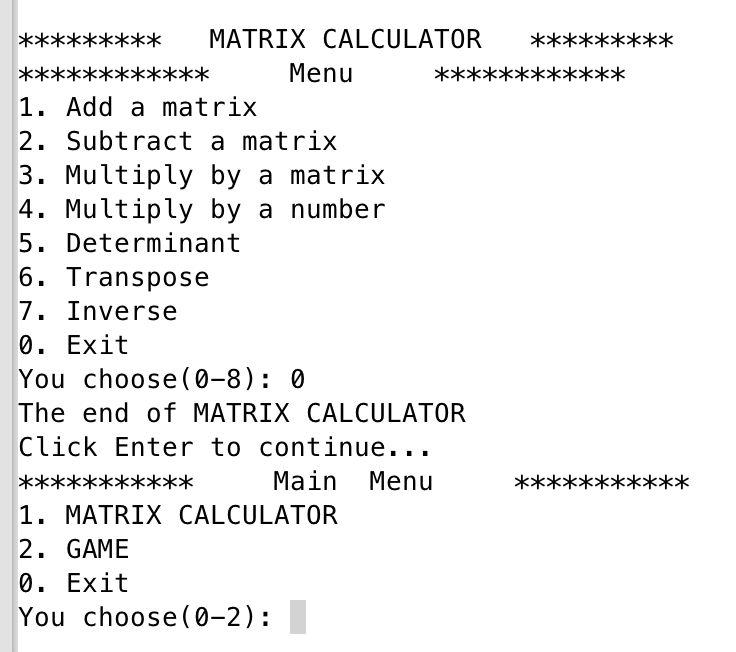
1. Input two matrices with same size, and it will output the result



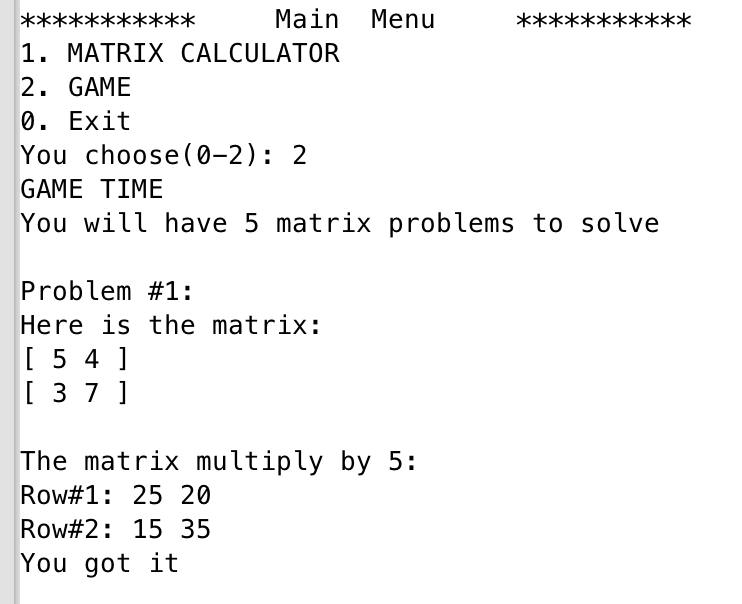
1. Choose 3 to multiply matrix by matrix



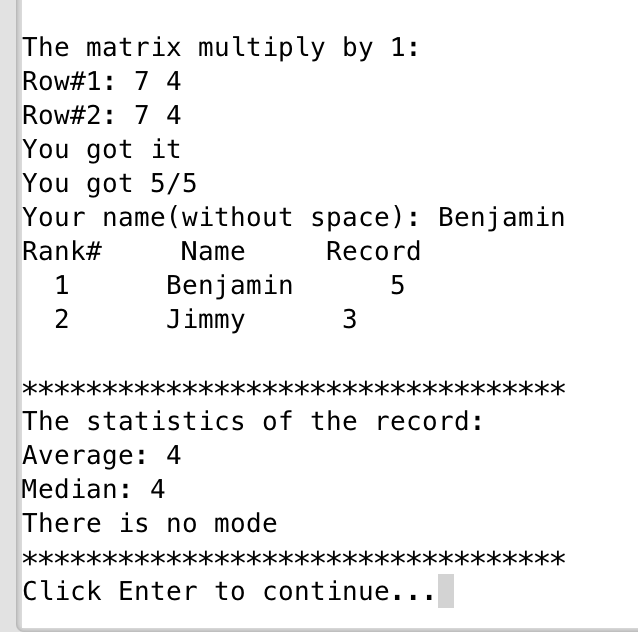
1. Choose 0 to exit to calculator



1. Play the Matrix game



1. After answer 5 problems, output the record



1. **Libraries**

**a. System libraries**

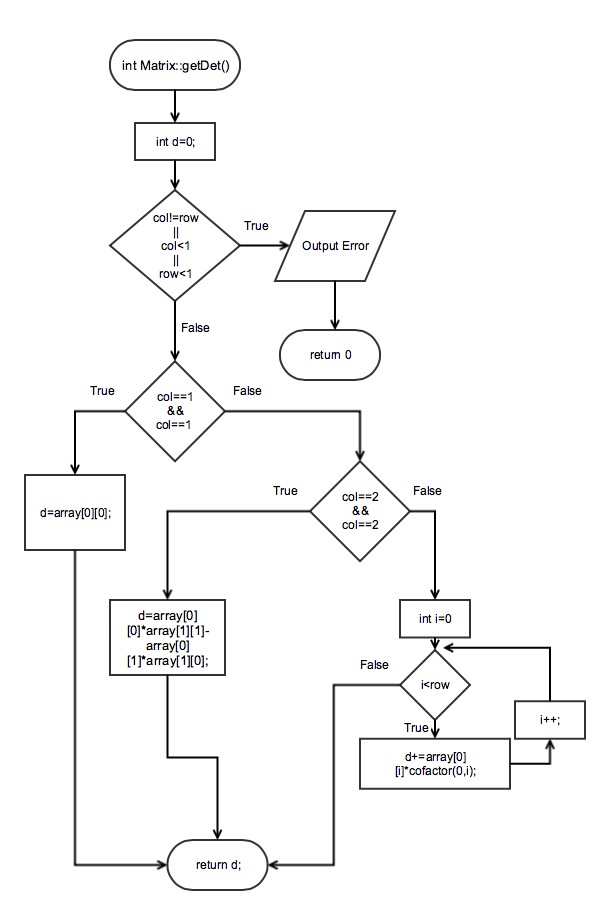
* **<** **#include <iostream>**
* **#include <string>**
* **#include <vector>**
* **#include <fstream>**
* **#include <set>**
* **#include <map>**
* **#include <algorithm>**
* **#include <iterator>**
* **b. User libraries**
* **#include "Matrix.h"**
* **#include "LnkList.h"**
* **#include "Record.h"**

1. **Concept covered**

|  |  |  |  |
| --- | --- | --- | --- |
| Concept | Type | Code | Location(line) |
| Linked List | LnkList<T> | LnkList<string> name; | 23 in Record.h |
| File i/o | fstream | fstream inFile("Record.txt",ios::in); | 125 in Record.cpp |
| Sort |  | void quickSort(LnkList<string> &,LnkList<int> &,int,int); | 27 in Record.h |
| Template function | template <class T> | template <class T>  void swap(T &a,T &b) | 31 in Record.h |
| Constructor |  | Matrix(int,int,int \*\*,string); | 30 Record.h |
| Destructor |  | ~Matrix(); | 32 Record.h |
| 2-d array | int \*\* | int \*\*array; | 19 Record.h |
| STL <vector> | vector<int> | vector<int> vec; | 21 Record.cpp |
| STL <set> | set<int> | set<int> s; | 22 Record.cpp |
| Iterator | set<int> :: iterator | set<int>::iterator pos; | 23 Record.cpp |
| STL<map> | map<int,int> | map<int,int> m; | 24 Record.cpp |
| Recursion |  | In the determinant function, call determinant(temp,tRow) | 342 Matrix.cpp |
| Recursion |  | In the Quick Sort | 101 Record.cpp |
| Operator Overloading |  | void operator=(const Matrix &right); | 54 in Matrix.h |

|  |  |  |  |
| --- | --- | --- | --- |
| Sort | Quick Sort | void Record::quickSort(LnkList<string> &nm, LnkList<int> &cr, int beg, int end) {  int p;  if(beg<end) {  p=pivot(nm,cr,beg,end);  quickSort(nm,cr,beg,p-1);  quickSort(nm,cr,p+1,end);  }  } | 97 Record.cpp |
| Operator Overloading |  | void operator=(const Matrix &right); | 54 in Matrix.h |

1. **Flowchart (A function beginning at Line#290 in Matrix.cpp)**

****