

HW4: Struct/Matrix & C++ 視覺化

Name	Student ID	Date	Grades
楊英豪	B10803207	2022/10/16	

Code:

```
#include <vector>
#include <iostream>
#include <time.h>
using namespace std;

// A type alias declaration of a vector of double value
using RealVec=vector<double>;

// A type alias for vector dimensions (size)
using Dim= RealVec::size_type;

//A type alias for vector type (e.g. double)
using DataType=RealVec::value_type;

//A type alias for a vector of vectores with doubles= a real matrix
using RealMatrix=vector<RealVec>;

void nonDiagonalSum(RealMatrix m)
{
    //Accumulate elements except the diagonal
    double non_diagonal_sum {};

    for (int r=0; r < m.size();++r)
        for(int c=0;c<m[0].size();++c)
            if (c != r)
                non_diagonal_sum += m[r][c];
    cout<<"Sum of m except the diagonal "<<non_diagonal_sum<<endl;
}

void generateMatrix(vector<vector<int>>&Matrix,int num_rows, int num_cols){

    for(auto it_row = Matrix.begin();it_row != Matrix.end();it_row++){
        for(auto it_col = it_row->begin(); it_col != it_row->end(); it_col++){
            *it_col = rand() %(201)-100;
        }
    }
}

void matrixAddition(vector<vector<int>>&M1, vector<vector<int>>&M2,
vector<vector<int>>&MPlus,int num_rows,int num_cols){
    for(int i = 0; i < num_rows; ++i){
```

```

        for(int j = 0; j < num_cols; ++j){
            MPlus[i][j] = M1[i][j] + M2[i][j];
        }
    }
}

void matrixSubstraction(vector<vector<int>>&M1, vector<vector<int>>&M2,
vector<vector<int>>&MMinus,int num_rows,int num_cols){
    for(int i = 0; i < num_rows; ++i){
        for(int j = 0; j < num_cols; ++j){
            MMinus[i][j] = M1[i][j] - M2[i][j];
        }
    }
}

void printMatrix(vector<vector<int>>&Matrix,int row, int col){
    for (auto it_row = Matrix.begin(); it_row != Matrix.end(); it_row++){
        for (auto it_col = it_row->begin(); it_col != it_row->end(); it_col++){
            cout << *it_col << " ";
        }
        cout << endl;
    }

// void printStructMatrix(){

// }

}

int rowAvg(vector<vector<int>>&M1, int& rand_row){
    int i,j;
    int sum = 0;
    for (i = rand_row-1; i < rand_row; ++i) {
        for (j = 0; j < 4; ++j) {
            // Add the element
            sum += M1[i][j];
        }
    }
    // cout<<rand_row<<endl;
    // cout<<sum;
    return sum;
}

struct HorzRow{
    int rand_row;
    vector<vector<int>>vec;
    int rowAvg;
};

int main(){
    // Using RealMatrix as it is
    RealMatrix m ={
        { 2, 2, 0, 11 },
        { 3, 4, 5, 0 },

```

```

        { -1, 2, -1, 7 }
    };

//Add next row
m.push_back({ 5, 3, 5, -3 });
    srand(time(0));
nonDiagonalSum(m);

int num_rows=4;
int num_cols=4;
vector<vector<int>>M1(num_rows,vector<int>(num_cols,0));
vector<vector<int>>M2(num_rows,vector<int>(num_cols,0));
vector<vector<int>>MPlus(num_rows,vector<int>(num_cols,0));
vector<vector<int>>MMinus(num_rows,vector<int>(num_cols,0));

cout<<"\nFirst Matrix M1"<<endl;
generateMatrix(M1,num_rows, num_cols);
printMatrix(M1,num_rows, num_cols);

cout<<"\nSecond Matrix M2"<<endl;
generateMatrix(M2,num_rows, num_cols);
printMatrix(M2,num_rows, num_cols);

cout<<"\nAddition"<<endl;
matrixAddition(M1,M2,MPlus,num_rows,num_cols);
printMatrix(MPlus,num_rows, num_cols);

cout<<"\nSubstraction"<<endl;
matrixSubstraction(M1,M2,MMinus,num_rows,num_cols);
printMatrix(MMinus,num_rows, num_cols);

int rand_row=(rand()%4)+1;

HorzRow horzRow=
{
    rand_row,
    M1,
    rowAvg(M1,rand_row),
};
cout<<"\nRow ID: "<<horzRow.rand_row<<endl;

for (int i = 0; i < horzRow.vec.size(); i++)
{
    for (int j = 0; j < horzRow.vec[i].size(); j++)
    {
        cout << horzRow.vec[i][j] << " ";
    }
    cout << endl;
}
cout<<"Average of Row:"<<horzRow.rowAvg<<endl;

}

```

```
[Running] cd "c:\Users\Reinaldo yang\OneDrive\Documents\Engineering Practices with modern programming skills\HW4\" && g++ main.cpp -o main && "c:\Users\Reinaldo yang\OneDrive\Documents\Engineering Practices with modern programming skills\HW4\main
Sum of m except the diagonal 42

First Matrix M1
60 100 67 18
-95 32 16 -18
41 -40 58 -31
-15 -34 -6 96

Second Matrix M2
32 18 50 -65
3 -96 32 44
5 90 22 65
-69 58 0 -1

Addition
92 118 117 -47
-53 -64 88 26
46 50 80 34
-84 24 -6 95

Subtraction
28 82 17 83
-98 128 -16 -62
36 -130 96 -96
54 -92 -6 97

Row ID: 2
60 100 67 18
-95 32 16 -18
41 -40 58 -31
-15 -34 -6 96
Average of Row: 65

[Done] exited with code=0 in 0.548 seconds
```

Part 2

Output:

```
ComplexTest.txt x +
1 |a = 100.01 200.02, b = 10002.2 -2000.22
2 a + b = 10102.2 -1800.2
3 a - b = -9902.21 2200.24
4 a * b = 1.40041e+06 1.8006e+06
5 a / b = 0.005769 0.0211512
6
7 a = -612.759 102.645, b = 250 14
8 a + b = -362.759 116.645
9 a - b = -862.759 88.645
10 a * b = -154627 17082.6
11 a / b = -2.42045 0.546125
12
13

>_ Console x Shell x +
> sh -c make -s
> ./main
10 20
10 20
Write re, press Enter, then im, press Enter
100.0005
102.2831
( 100.001 102.283 ) + ( 100.001 102.283 ) = ( 200.001 204.566 )
( 100.001 102.283 ) + ( 5 0 ) = ( 105.001 102.283 )
Enter 1st complex
-612.7592
102.645
Enter 2nd complex
250.0 14
>
> ls
Complex.cpp      main             'main(ORG)-cpp.txt'
Complex.h        main.cpp         Makefile
ComplexTest.cpp  main-debug      replit.nix
ComplexTest.txt  main.o
> []
```