Data Structure HW3

start date = 9 월 25 일 (금)
due date = 10 월 9 일 (금) 23시 59분 59초

submit 방법 : submit pem_ta hw3a or hw3b (1분반 : a, 2분반 : b)
※ 제출 시 실행파일을 지우고 제출합니다.
※ 지정한 파일명과 다를 경우 0점 처리합니다.

1 makefile

cat makefile

```
hw3a: hw3a.o matrixa.o
	g++ -o hw3a hw3a.o matrixa.o
hw3a.o matrixa.o: matrixa.h
hw3b: hw3b.o matrixb.o
	g++ -o hw3b hw3b.o matrixb.o
hw3b.o matrixb.o: matrixb.h
```

2 행렬 연산의 구현

2.1 다음과 같이 동작하는 프로그램을 작성하려 한다.

Matrix1 Determinant : -2

Matrix2 Determinant : 0

==========

Matrix Transpose

==========

1 3

2 4

==========

Matrix Add

==========

2 3

4 5

==========

Matrix Sub

==========

0 1

2 3

==========

Matrix Multi

==========

3 3

7 7

2.2 main 프로그램 (hw3a.cpp)은 다음과 같다.

```
#include "matrixa.h"
#include <iostream>
using namespace std;
int main(){
   Matrix matrix1(1,2,3,4);
   Matrix matrix2(1,1,1,1);
   Matrix matrix3;
   cout << "Matrix1 Determinant : " << matrix1.GetDet() << endl;
   cout << "Matrix2 Determinant : " << matrix2.GetDet() << endl;
   cout << "==========" << endl;
   cout << "Matrix Traspose" << endl;
   cout << "==========" << endl;
   matrix1.Transpose();
   matrix1.ShowMatrix();
   matrix1.Transpose();
   cout << "Matrix Add" << endl;
   cout << "==========" << endl;
   matrix3=matrix1+matrix2;
   matrix3.ShowMatrix();
   cout << "==========" << endl;
   cout << "Matrix Sub" << endl;
   cout << "==========" << endl;
   matrix3=matrix1-matrix2;
   matrix3.ShowMatrix();
   cout << "==========" << endl;
   cout << "Matrix Multi" << endl;
   cout << "==========" << endl;
   matrix3=matrix1*matrix2;
   matrix3.ShowMatrix();
   return 0;
}
```

2.3 다음을 matrixa.h로 지정하라.

```
#ifndef MATRIX_H
#define MATRIX_H
#include <iostream>
using namespace std;
class Matrix{
public:
   Matrix(int a=0, int b=0, int c=0, int d=0);
   ~Matrix() {}
   void ShowMatrix();
   void Transpose();
   int GetDet();
   Matrix operator+(const Matrix& a);
   Matrix operator-(const Matrix& a);
   Matrix operator*(const Matrix& a);
   void operator=(const Matrix& a);
private:
   int m[2][2];
};
#endif
```

2.4 다음 matrixa.cpp를 완성하라.

```
#include "matrixa.h"
Matrix::Matrix(int a, int b, int c, int d){
//....
}
void Matrix::Transpose(){
//....
}
Matrix Matrix::operator+(const Matrix &a){
//....
}
Matrix Matrix::operator-(const Matrix &a){
//....
}
Matrix Matrix::operator*(const Matrix &a){
//....
}
void Matrix::operator=(const Matrix &a){
//....
}
void Matrix::ShowMatrix() {
//....
}
int Matrix::GetDet(){
//....
}
```

3 3 * 3 행렬연산의 구현

3.1 위의 실습과 마찬가지로 다음을 구현하시오.

hw3b.cpp matrixb.cpp matrixb.h

3.2 다음의 3 * 3 행렬에 대하여 2번과 같은 연산을 적용하시오.

$$Matrix1 = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

$$Matrix2 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

4 보고서 작성

- 연산자 오버로딩의 개념과 구현방법 설명
- matrix.h 와 matrix.cpp의 차이점에 대해 기술
- 실습하면서 어려웠던 점이 있었다면 기술

5 주의사항

- make 명령어를 통해 실행할 예정이므로, 파일이름을 정확히 해야한다.
- Cheating 적발시 F.
- 메일로 제출된 과제는 무시한다.

6 질문

- pemta806@gmail.com 으로 질문한다.
- 질문은 최대한 자세하게 한다.
- 코드 첨부가 필요하다면 이지미가 아닌 텍스트로 첨부한다.