**Report Template:**

**CS205 C/ C++ Program Design Assignment 1**

**Name**: 罗森滋

**SID**: 3120001095

Part 1. Source Code

#include<iostream>

class Matrix // class declaration

{

private:

int rows;

int col;

float\*\* elem;

public:

void scale(int x, int y);

void input(int, int);

void value();

Matrix operator\*(const Matrix& m) const;

Matrix operator+(const Matrix& m) const;

Matrix operator=(const Matrix& m) const;

void show();

};

int main() {

Matrix A, B,C;

int x0, y0, x1, y1;

std::cout << "Please input the scale of MatirxA:";

std::cin >> x0>>y0;

A.scale(x0, y0);

A.input(x0, y0);

std::cout << "Please input the data of MatrixA:" << std::endl;

A.value();

std::cout << "Please input the scale of MatirxB:";

std::cin >> x1>>y1;

B.scale(x1, y1);

B.input(x0, y0);

std::cout << "Please input the data of MatrixB:" << std::endl;

B.value();

C = A.operator+(B);

std::cout << "A+B=" <<std::endl;

C.show();

A.operator=(B);

std::cout << "A=B" << std::endl;

std::cout << "A is:"<<std::endl;

A.show();

std::cout << "B is" << std::endl;

B.show();

std::cout << "A\*B=" << std::endl;

C=A.operator=(B);

C.show();

A.~Matrix();

B.~Matrix();

return 0;

}

void Matrix::scale(int x, int y) {

elem = new float\* [x];

for (int i = 0; i < x; i++) {

elem[i] = new float[y];

if (elem[i][y] == 'NULL') {

std::cout << "Invalid value,please input again" << std::endl;

return;

}

}

return;

}

void Matrix::input(int x, int y) {

rows = x;

col = y;

return;

}

Matrix Matrix::operator+(const Matrix& m) const {

if (rows != m.rows && col != m.rows) {

std::cout << "ERROR!" << std::endl;

return \*this;

}

Matrix C;

C.scale(rows, col);

for (int i = 0; i < col; i++) {

for (int j = 0; j < rows; j++) {

C.elem[i][j] = elem[i][j] + m.elem[i][j];

}

}

return C;

}

Matrix Matrix::operator\*(const Matrix& m) const {

if (rows != m.col) {

std::cout << "ERROR!" << std::endl;

return \*this;

}

Matrix C;

C.scale(rows, col);

for (int i = 0; i <= col; i++) {

for (int j = 0; j <= rows; j++) {

for (int k = 0; k <= rows; k++)

C.elem[i][j] = (elem[i][k]) \* (m.elem[k][j]);

}

}

return C;

}

Matrix Matrix::operator=(const Matrix& m) const {

if (rows != m.col && col != m.rows) {

std::cout << "ERROR!" << std::endl;

return \*this;

}

for (int i = 0; i < col; i++) {

for (int j = 0; j < rows; j++) {

elem[i][j] = m.elem[i][j];

}

}

return \*this;

}

void Matrix::show() {

for (int i = 0; i < col; i++) {

for (int j = 0; j < rows; j++) {

std::cout << elem[i][j] << " ";

}

}

return;

}

void Matrix::value() {

for (int i = 0; i < col; i++) {

for (int j = 0; j < rows; j++) {

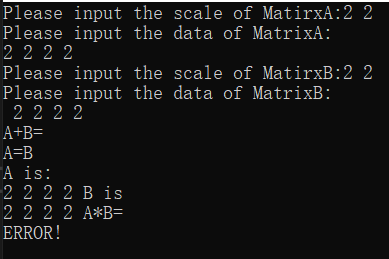
std::cin >> elem[i][j];

}

}

return;

}



Part 3. Difficulties & Solutions, or others

# In this program,I find when the function return a class ,like Matric C,and it will go to destructor and crash the program.