ОТЧЕТ

о лабораторной работе №2

по теме: Одномерные и двумерные массивы

по дисциплине: Разработка программных модулей

09.02.07 Информационные системы и программирование

УНИВЕРСИТЕТ ИТМО

Факультет среднего профессионального образования

Проверил:

\_\_\_\_\_\_\_\_\_Антонов М.Б.

Дата: «\_\_» \_\_\_\_\_\_\_ 20 г.

Оценка:

Выполнил:

Студент группы Y2333

\_\_\_\_\_\_\_\_ Харченко Д.И.

Санкт-Петербург 2019

**Цель работы:** создать программу работающую с двумерными массивами.

**Задачи:**

* научиться работать с двумерными массивами;
* научиться передавать и возвращать несколько значений функции;
* научиться создавать динамические массивы.

**Задание:**

Пользователь должен задать размер двух матриц. Ввод с клавиатуры:

* количество строк первой матрицы;
* количество столбцов первой матрицы;
* количество строк второй матрицы;
* количество столбцов второй матрицы.

Требуется реализовать обязательно в виде отдельных функций:

* сумму двух матриц;
* произведение двух матриц;
* транспонирование матрицы;
* сложение и умножение матрицы на число.

Если какое-либо действие невозможно выполнить — написать соответствующее сообщение об ошибке.

**Текст программы**:

#include <iostream> //объявление библиотеки

#include <locale.h> // позволяет использовать русский язык

#include <cstdlib> // содержит srand() и rand()

using namespace std; //обявление пространства имён std

void FillArray(int\*\* array, int len,int wid){//ф-ция,заполняющая массив

for (int i = 0; i < len; i++){//итерируемся по строкам

array[i] = new int[wid];//выделяем дин. память

for (int j = 0; j < wid; j++){//итерируемся по столбцам

array[i][j] = rand()%9+1;//заполн. Ранд. значениями

}}}//закрываем циклы и ф-цию

template <typename A>//используем шаблон

void PrintMas(A \*\*a, const int& n1, const int& m1){//ф-ция отвечающая за вывод массива

for(int i = 0; i < n1; i++){//итерируемся по строкам

for(int j = 0; j < m1; j++){//итерируемся по столбцам

if(a[i][j] < 10){//проверяем сколько места потребуется

cout.width(2);//форматируем

cout << a[i][j] << " ";}//выводим

if(a[i][j] >= 10)//проверяем сколько места потребуется

cout.width(1); //форматируем

cout << a[i][j] << " ";}//выводим

}

cout << endl; //новая строка

}//закрываем цикл

}//закрываем ф-цию

void SumOfMatrix(int \*\*a, int \*\*b,int \*\*c,int& len1, int& wid1){ //сумма

for(int i = 0; i < len1; i++){//итерируемся по строкам

for(int j = 0; j < wid1; j++){//итерируемся по столбцам

c[i][j] = a[i][j] + b[i][j]; //скаладываем элементы в новую матрицу

}//закрываем цикл

}закрываем цикл

}закрываем ф-цию

void MultiplyOfMatrix(int \*\*a, int \*\*b, int \*\*d,int width, int height){ //умножение

for(int i = 0; i < width; i ++ ){ //итерируемся по столбцам

for(int j = 0; j < height; j++){//итерируемся по строкам

int val = 0; //новая переменная

for(int v = 0; v < height; v++){//проходим по строкам

val += a[i][v]\*b[v][j]; //суммируем

}//закрываем внутренний цикл

d[i][j] = val; //заполняем новый массив

}//закрываем внутренний цикл

}//закрываем внешний цикл

}//закрываем ф-цию

int\*\* Transp(int\*\* array,int n, int m) //ф-ция транспонирования

{ int i,j; //инициализивкем счетчики

int \*\*b; //новый массив

b=new int \*[n]; //выделение дин. памяти

for (i = 0 ;i < n ; i++){// n = length; m = width

b[i] = new int[m]; //дин. Выделение памяти

if(b[i] == NULL){ //проверка на пустоту

cout<< "Error" << endl; //вывод сообщение об ошибке

exit(1);}} //выход из программы

for (i = 0 ; i < n ; i++){//итерируемся по строкам

for ( j = 0 ; j < m ; j ++){//итерируемся по столбцам

b[j][i] = array[i][j]; //переписываем матрицу

}//закрываем внешний цикл

}//закрываем внутренний цикл

return b; //возращаем переделанную матрицу

}//закрываем ф-цию

double\*\* Normalisation(int\*\* arr,int n, int m) //ф-ция нормализации матрицы

{ int i,j; //инициализация счетчиков

double \*\*b; //объявляем новый массив

b=new double \*[n]; //выд. Дин. памяти

for (i = 0; i < n; i++){//итерируемся по строкам

b[i] = new int[m]; //выд. Дин. памяти

double c = 0; //сюда будет записан элемент главной диагонали(для каждой строки)

for ( j = 0 ; j < m ; j ++){//итерируемся по столбцам

if(i == j){ //проверка на принадлежность к гл. диа.

c = arr[i][j]; //присв. Временного знач-ния

b[i][j] = arr[i][j]/c; //нормализация

}}//конец усл и цикла столбцов

c = 0; //освежение элемента гл. диа

}//конец цикла строк

return b; //возращаем нормализованную матрицу

}//конец ф-ции

double Determ(int\*\* Arr, int size) //ф-ция определителя

{

int i = 0,j = 0; //инициализируем счетчики

double det=0; //значение определителя

int\*\* matr; //вспомогательная матрица

if(size==1) //если состоит из одного элемента

{

det=Arr[0][0]; //то опр равен самому эл-ту

}

else if(size==2) //если из двух эл-тов

{

det=Arr[0][0]\*Arr[1][1]-Arr[0][1]\*Arr[1][0];//перемножаем крестиком

}

else//если размер матрицы больше

{

matr=new int\*[size-1]; //определяем размер вспомогательной матрицы

for(i=0;i<size;++i) //итерируемся по строкам

{

for(j=0;j<size-1;++j) //итерируемся по столбцам кроме последнего

{

if(j<i) //определяем откуда элемент

matr[j]=Arr[j]; //присваиваем вспомогательной матрице

else

matr[j]=Arr[j+1]; //присваиваем вспомогательной матрице

}

det+=pow((double)-1, (i+j))\*Determ(matr, size-1)\*Arr[i][size-1]; //вычисляем определитель

}

delete[] matr; //удаляем больше не нужный массив

}

return det; //возвращаем результат из ф-ции

}

int\*\* SumWithUsersNumber(int\*\* arr,int n, int m) //ф-ция суммирующая каждый элемент матрицы с числом введеным пользователем

{ int i, j, c; //объявление переменных

int \*\*b; //объявление массива

cout << "Enter number with which you would sum matrix" << endl; //вывод строковой константы

cin >> c; //считываем переменную

b=new int \*[n]; //объявление массива

for (i = 0; i < n; i++){// проходим по каждой строке

b[i] = new int[m]; //объявляем массив

for ( j = 0 ; j < m ; j ++){//проходим по каждому столбцу

b[i][j] = arr[i][j]+c; //прибавление каждому элементу

}

}

return b; //возвращаем матрицу

}

int\*\* MultiplyWithUsersNumber(int\*\* arr,int n, int m) //ф-ция перемножающая каждый элемент матрицы с числом введеным пользователем

{ int i, j, c; //объявление переменных

int \*\*b; //объявление массива

cout << "Enter number with which you would sum matrix" << endl; //вывод строковой константы

cin >> c; //считываем переменную

b=new int \*[n]; //объявление массива

for (i = 0; i < n; i++){// проходим по каждой строке

b[i] = new int[m]; //объявляем массив

for ( j = 0 ; j < m ; j ++){//проходим по каждому столбцу b[i][j] = arr[i][j]\*c;//умножение элемента на число

}

}

return b; //возвращаем массив

}

int main(){

int len1, wid1, len2, wid2;//ширина и длина обеих матриц

cout << "Enter width of first matrix" << endl; //вывод строковой константы

cin >> wid1;//ввод ширины первой матрицы

cout << "Enter length of first matrix" << endl; //вывод строковой константы

cin >> len1; //ввод длины первой матрицы

int \*\*a; //объявление массива

a = new int\*[len1]; //объявление массива

FillArray(a,len1,wid1); //вызов ф-ции заполнения масива

cout << "First matrix" << endl;

PrintMas(a, len1, wid1); //вызов ф-ции распечатки массива

cout << "Enter width of second matrix" << endl;

cin >> wid2; //ввод ширины второй матрицы

cout << "Enter length of second matrix" << endl;

cin >> len2; //ввод длины второй матрицы

int \*\*b; //объявление второго массива

b = new int\*[len2]; //объявление второго массива

FillArray(b,len2,wid2); //заполн. Второго массива

cout << "Second matrix" << endl;

PrintMas(b, len2, wid2); //выводим второй массив

//сумма

if ( (len1 == len2) && (wid1 == wid2)){ //проверка можно ли складывать матрицы

cout << "Sum of introduced matrixes is possible" << endl;

int \*\*c; //создаем новую матрицу

c = new int\*[len1]; //создаем новую матрицу

for (int i = 0; i < len1 ;i ++){ //итерируемся по ней

c[i] = new int[wid1]; //создаем новую матрицу

}

SumOfMatrix(a, b,c, len1, wid1); //вызываем ф-цию сложения

cout << "Ladies and Gentlemen, present to you sum of first and second matrix is" << endl; //вывод строковой константы

PrintMas(c, len1, wid1); //распечатываем получившуюся матрицу

}else{ //в случае если матрицы разных размеров

cout << "Ladies and Gentlemen, sum of introduced matrixes is impossible" << endl; //вывод строковой константы

}

//умножение

if((wid1 == len2)||(wid2 == len1)){ //проверяем возможно ли умножение

cout << "multiplication of introduced matrixes is possible" << endl; //вывод строковой константы

int \*\*d; //создаем новый массив

if(wid1 == len2){ //в случае когда число столбцов первой матрицы равно числу строк второй

d = new int\*[len2]; //определяем размер

for (int i = 0; i < len2 ;i ++){ //итерируемся по строкам

d[i] = new int[wid1];} //создаем новый массив

MultiplyOfMatrix(a, b, d , wid1,len2); //вывзываем ф-цию перемножения

cout << "multiplication of introduced matrixes is "<< endl; //вывод строковой константы

PrintMas(d, len2, wid1); //распечатываем результат

}else{ //в случае когда число строк первой матрицы равно числу столбцов второй

d = new int\*[len1]; //создаем новый массив

for (int i = 0; i < len1 ;i ++){ //итерируемся по нему

d[i] = new int[wid2];} //создаем новый массив

MultiplyOfMatrix(a, b, d , wid2 ,len1); //вызываем ф-цию умножения

cout << "multiplication of introduced matrixes is " << endl;

PrintMas(d, len1, wid2);} //распечатываем результат

}else{ //если матрицы невозможно умножить

cout << "multiplication of introduced matrixes is impossible" << endl;

}

//транспонирование

int \*\*transp1; //создаем результирующую матрицу

transp1 = Transp(a, len1, wid1); //вызываем ф-цию транспонирования

cout << "Ladies and Gentlemen, present to you the first transported matrix" << endl; //вывод строковой константы

PrintMas(transp1, wid1, len1); //распечатываем результат

int \*\*transp2; //создаем вспомогательную матрицу

transp2 = Transp(b, len2, wid2); //присваиваем результат ф-ции

cout << "Ladies and Gentlemen, present to you the second transported matrix" << endl;вывод строковой константы

PrintMas(transp2, wid2, len2); //распечатываем результат

//нормализовать

double \*\*norm1; //результативная матрица

norm1 = Normalisation(a, len1, wid1); //вызываем ф-цию транспонирования

cout << "Ladies and Gentlemen, present to you the first normalized matrix" << endl; //выводим результат

PrintMas(norm1, wid1, len1); //выводим результат

double \*\*norm2; //вторая результативная матрица

norm2 = Normalisation(b, len2, wid2); //вызываем ф-цию странспонирования

cout << "Ladies and Gentlemen, present to you the second normalized matrix" << endl; //вывод строковой константы

PrintMas(norm2, wid2, len2); //распечатываем результат

// определитель

if ( len1 == wid1 ){ //проверка на квадратность первой матрицы

cout << "Determinant of first matrix is " << Determ(a,len1)<< endl; //вывод строковой константы

}else{//в случае если она прямоугольная

cout << "Impossiple to calculate first determinant "<< endl; //вывод строковой константы

}

if ( len2 == wid2 ){ //проверка на квадратность второй матрицы

cout << "Determinant of second matrix is " << Determ(b,len2)<< endl; //вывод строковой константы

}else{//в лучае если матрица вторая

cout << "Impossiple to calculate second determinant "<< endl; //вывод строковой константы

}

//сложение с числом пользователя

int \*\*sum\_users\_number1;// создаем новый массив

sum\_users\_number1 = SumWithUsersNumber(a, len1, wid1); //вызов функции

cout << "First Matrix summed with user's number is" << endl; //вывод строковой константы

PrintMas(sum\_users\_number1, len1, wid1); //распечатываем результат

int \*\*sum\_users\_number2; //создаем масив для записи

sum\_users\_number2 = SumWithUsersNumber(b, len2, wid2); //вызов ф-ции

cout << "Second Matrix summed with user's number is" << endl; //вывод строковой константы

PrintMas(sum\_users\_number2, len2, wid2); //распечатываем массив

//умножение с числом пользователя

int \*\*multiply\_users\_number1; //создаем массив

multiply\_users\_number1 = MultiplyWithUsersNumber(a, len1, wid1); //вызов ф-ции

cout << "First matrix mulplied with user's number is" << endl; //вывод строковой константы

PrintMas(multiply\_users\_number1, len1, wid1); //распечатываем массив

int \*\*multiply\_users\_number2; //создаем массив

multiply\_users\_number2 = MultiplyWithUsersNumber(b, len2, wid2); //вызов ф-ции

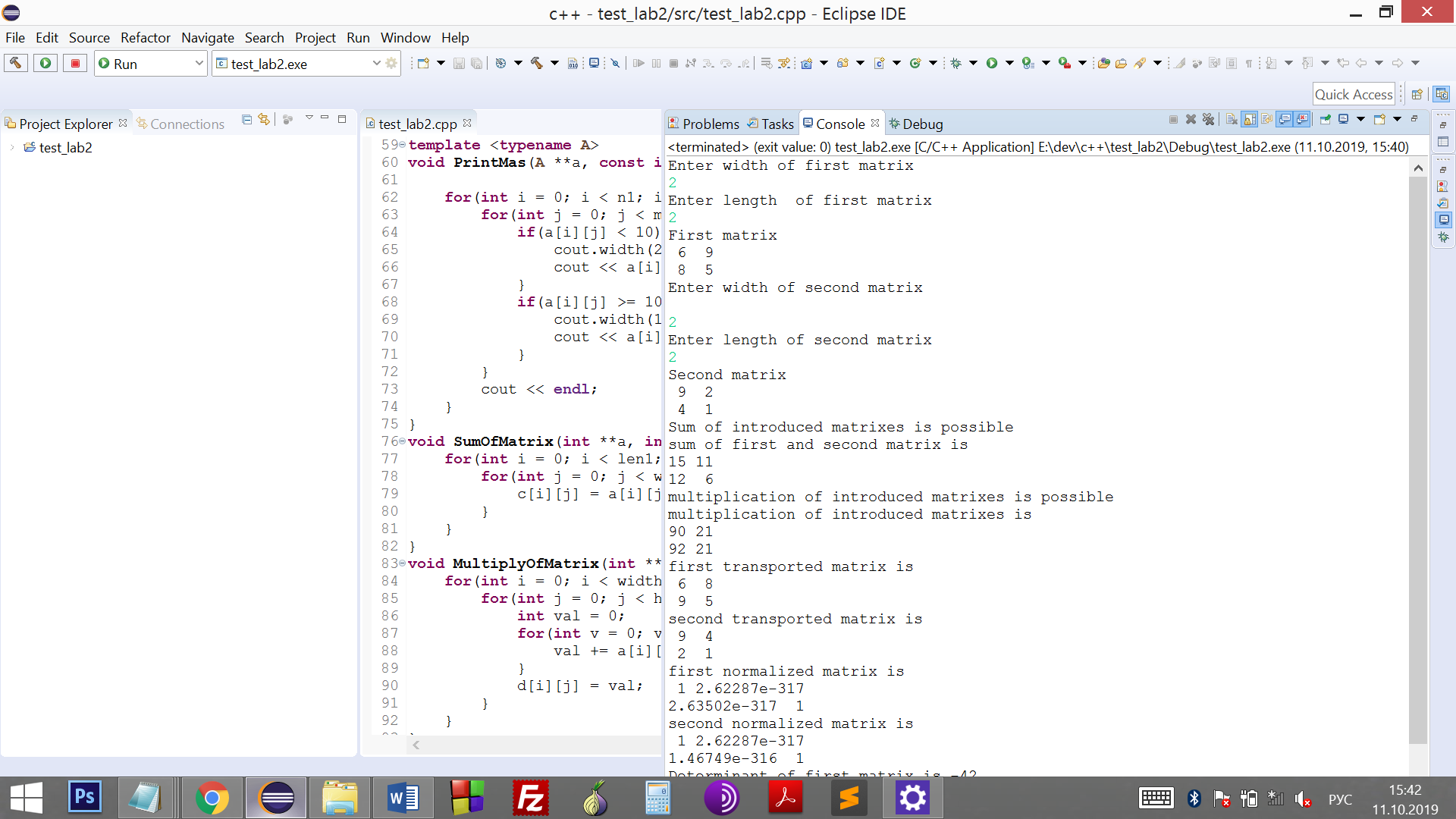
cout << "Second matrix mulplied with user's number is" << endl; //вывод строковой константы

PrintMas(multiply\_users\_number2, len2, wid2); //распечатываем массив

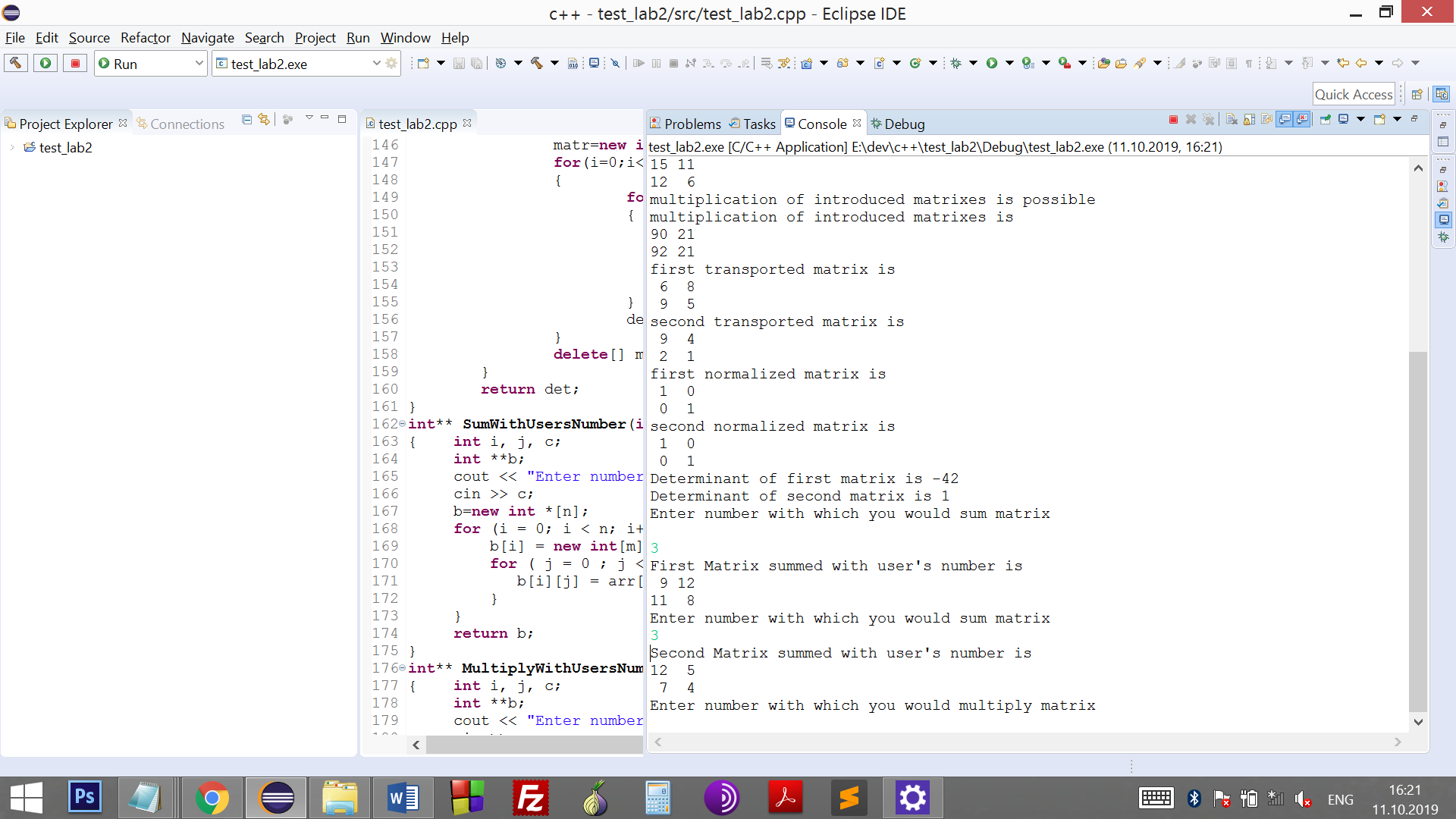
return 0;

}

Протокол работы программы представлен на рисунках 1 и 2.



*Рисунок 1- Протокол работы программы*



*Рисунок 2- Протокол работы программы*