Министерство образования и науки Российской Федерации Федеральное агентство по образованию

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Вятский государственный университет»

Факультет автоматики и вычислительной техники

Кафедра электронных вычислительных машин

Отчет по лабораторной работе №1 дисциплины

«Технологии программирования»

Выполнил студент группы ИВТ-22 /Крючков И. С/ Проверил /Долженкова М. Л./

Киров 2022

1. Задание

Написать программу, реализующую умножение матриц, используя числа типов byte и float с использованием перегруженных функций и шаблонов.

1. Листинг программы

#include <iostream>

#include <limits>

using namespace std;

void set\_matrix(char \*\*&, char \*\*&);

void set\_matrix(float \*\*&, float \*\*&);

void get\_value\_matrix(char &);

void get\_value\_matrix(float &);

int arows, acols, brows, bcols;

\_\_int16\*\* mx(char\*\*, char\*\*);

float\*\* mx(float\*\*, float\*\*);

float\*\* mx(float\*\* a, float\*\* b) {

float\*\* c;

c = new float \*[arows];

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

c[i] = new float[bcols];

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

c[i][j] = 0;

for (int k = 0; k < brows; k++) {

c[i][j] += a[i][k] \* b[k][j];

}

}

}

return c;

}

\_\_int16\*\* mx(char\*\* a, char\*\* b) {

\_\_int16\*\* c;

c = new \_\_int16\* [arows];

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

c[i] = new \_\_int16[bcols];

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

c[i][j] = 0;

for (int k = 0; k < brows; k++) {

c[i][j] += a[i][k] \* b[k][j];

}

}

}

return c;

}

void get\_value\_matrix\_size(int &value) {

int v;

while (true) {

if ((cin >> v).good()) {

if (v >= 1 and v <= 1000) {

value = v;

break;

}

else {

cout << "Invalid value" << endl;

cin.clear();

cin.ignore(numeric\_limits<streamsize>::max(), '\n');

continue;

}

}

else {

cout << "Invalid value" << endl;

cin.clear();

cin.ignore(numeric\_limits<streamsize>::max(), '\n');

continue;

}

}

cin.clear();

cin.ignore(numeric\_limits<streamsize>::max(), '\n');

}

void get\_value\_type(int &value) {

int v;

while (true) {

if ((cin >> v).good()) {

if (v >= 1 and v <= 3) {

value = v;

break;

}

else {

cout << "Invalid value" << endl;

cin.clear();

cin.ignore(numeric\_limits<streamsize>::max(), '\n');

continue;

}

}

else {

cout << "Invalid value" << endl;

cin.clear();

cin.ignore(numeric\_limits<streamsize>::max(), '\n');

continue;

}

}

cin.clear();

cin.ignore(numeric\_limits<streamsize>::max(), '\n');

}

void get\_value\_matrix(float &value) {

float v;

while (true) {

if (cin >> v) {

value = v;

cin.clear();

cin.ignore(numeric\_limits<streamsize>::max(), '\n');

break;

}

else {

cout << "Invalid value" << endl;

cin.clear();

cin.ignore(numeric\_limits<streamsize>::max(), '\n');

continue;

}

}

}

void get\_value\_matrix(char &value) {

char str[5];

int tv;

while (true) {

cin.getline(str, 5);

bool f = true;

bool z = false;

int k = 0;

while (k < 4) {

if (!(str[k] >= '0' and str[k] <= '9' or (str[k] == '-' and k == 0))) {

if (int(str[k]) == 0) {

if(k == 0){

f = false;

z = false;

cout << "Invalid value " << endl;

}

break;

}

else {

f = false;

if(cin.rdbuf()->in\_avail() >= 1){

z = true;

}

cout << "Invalid value " << endl;

break;

}

}

k++;

}

if (str[0] == '-' and k == 1 and f == true) {

f = false;

cout << "Invalid value " << endl;

}

if (f) {

tv = atoi(str);

if (tv >= -128 && tv <= 127) {

value = tv;

if (cin.rdbuf()->in\_avail() > 0) {

cin.clear();

cin.ignore(numeric\_limits<streamsize>::max(), '\n');

}

break;

}

else {

cout << "Invalid value " << endl;

if (cin.rdbuf()->in\_avail() > 0) {

cin.clear();

cin.ignore(numeric\_limits<streamsize>::max(), '\n');

}

}

}

if (z) {

cin.clear();

cin.ignore(numeric\_limits<streamsize>::max(), '\n');

}

}

}

void set\_matrix(char \*\*&a, char \*\*&b) {

a = new char \*[arows];

cout << "Input A matrix" << endl;

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

a[i] = new char[acols];

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

cout << "a[" << i << "][" << j << "] = ";

get\_value\_matrix(a[i][j]);

}

}

cout << "Matrix A" << endl;

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

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

cout << int(a[i][j]) << " ";

}

cout << endl;

}

b = new char \*[brows];

cout << "Input B matrix" << endl;

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

b[i] = new char[bcols];

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

cout << "b[" << i << "][" << j << "] = ";

get\_value\_matrix(b[i][j]);

}

}

cout << "Matrix B" << endl;

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

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

cout << int(b[i][j]) << " ";

}

cout << endl;

}

}

void set\_matrix(float \*\*&a, float \*\*&b) {

a = new float\* [arows];

cout << "Input A matrix" << endl;

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

a[i] = new float[acols];

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

cout << "a[" << i << "][" << j << "] = ";

get\_value\_matrix(a[i][j]);

}

}

cout << "Matrix A" << endl;

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

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

cout << a[i][j] << " ";

}

cout << endl;

}

b = new float \*[brows];

cout << "Input B matrix" << endl;

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

b[i] = new float[bcols];

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

cout << "b[" << i << "][" << j << "] = ";

get\_value\_matrix(b[i][j]);

}

}

cout << "Matrix B" << endl;

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

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

cout << b[i][j] << " ";

}

cout << endl;

}

}

template<class T>

void print\_result(T\*\* matrix) {

cout << "Result matrix" << endl;

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

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

if(typeid(T) != typeid(float)){

cout << int(matrix[i][j]) << " ";

}

else {

cout << matrix[i][j] << " ";

}

}

cout << endl;

}

}

int main()

{

setlocale(LC\_ALL, "Russian\_Russia.1251");

int n = 0;

do {

cout << "Matrix A rows (1-1000): " << endl;

get\_value\_matrix\_size(arows);

cout << "Matrix A cols (1-1000): " << endl;

get\_value\_matrix\_size(acols);

cout << "Matrix B rows (1-1000): " << endl;

get\_value\_matrix\_size(brows);

cout << "Matrix B cols (1-1000): " << endl;

get\_value\_matrix\_size(bcols);

if (acols != brows) {

cout << "Error: Matrix A cols != Matrix B rows" << endl;

}

} while (acols != brows);

cout << "Select type:" << endl

<< "1 - float" << endl

<< "2 - byte" << endl

<< "3 - Exit" << endl;

while (true) {

get\_value\_type(n);

if (n == 1) {

float \*\*a, \*\*b;

set\_matrix(a, b);

print\_result(mx(a, b));

break;

}

else if (n == 2) {

char \*\*a, \*\*b;

set\_matrix(a, b);

print\_result(mx(a, b));

break;

}

else if (n == 3) {

return 0;

}

else {

continue;

}

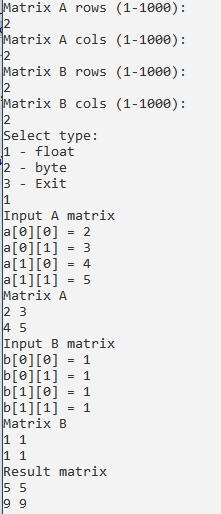
}

system("pause");

return 0;

}

1. Экранные формы



1. Вывод

В ходе выполнения лабораторной работы были получены необходимые знания синтаксиса языка программирования C++. Написана программа на языке C++, выполняющая умножения матриц, используя числа типов byte и float. Освоены принципы построения перегруженных и шаблонных функций, применены на практике при написании программы. Изучены принципы работы стандартных потоков ввода/вывода. Реализованы механизмы валидации входных данных.