Балтийский государственный технический университет «ВОЕНМЕХ» им. Д. Ф. Устинова

Кафедра И5  
«Информационные системы и программная инженерия»

**Лабораторная работа № 3** по дисциплине «Компьютерный практикум»на тему **«Создание динамических библиотек при помощи набора компиляторов и утилит GCC и их применение**.**»**

Выполнил:

Студент Селюхов М.М.  
 Группа И582

**Преподаватель**:  
 Вальштейн К. В.

Санкт-Петербург

2019

*Цель работы:*

Изучить процесс создания динамических библиотек при помощи набора компиляторов и утилит GCC и особенности их применения.

*Задание:*

В матрице А (7х6) и массиве В (67) заменить все отрицательные числа их квадратами.

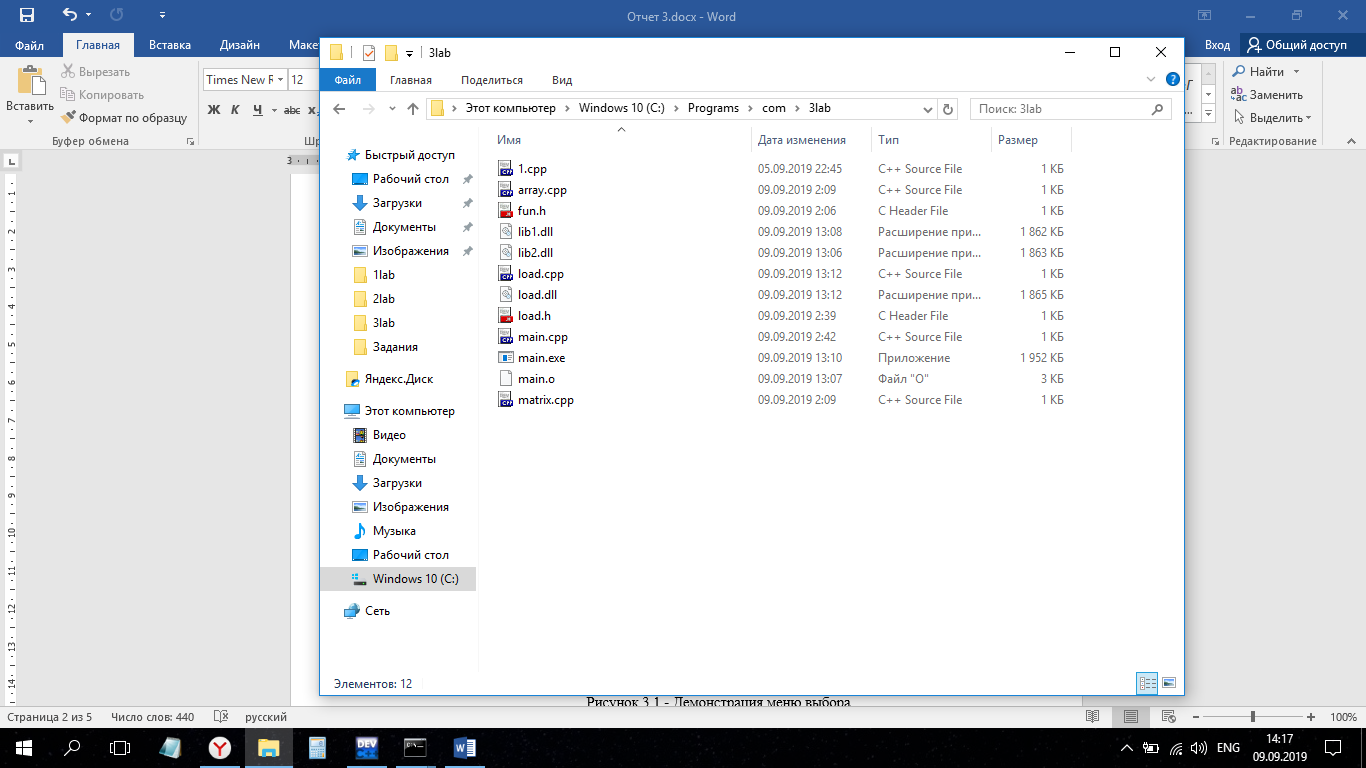
Путь созданных файлов:

Рисунок 3.1 - Демонстрация расположения файлов

Команды использованные для компиляции динамической библиотеки:

Создаем динамическую библиотеку из array.cpp:

C:\Programs>MinGW64\bin\g++ -fPIC -c -o com\3lab\array.o com\3lab\array.cpp

C:\Programs>MinGW64\bin\g++ -shared -o com\3lab\libarray.dll com\3lab\array.o

Создаем динамическую библиотеку из matrix.cpp:

C:\Programs>MinGW64\bin\g++ -fPIC -c -o com\3lab\matrix.o com\3lab\matrix.cpp

C:\Programs>MinGW64\bin\g++ -shared -o com\3lab\libmatrix.dll com\3lab\matrix.o

Создаем динамическую библиотеку из load.cpp:

C:\Programs>MinGW64\bin\g++ -fPIC -c -o com\3lab\load.o com\3lab\load.cpp

C:\Programs>MinGW64\bin\g++ -shared -o com\3lab\libload.dll com\3lab\load.o

Скомпилируем и запустим полученную программу:

C:\Programs>MinGW64\bin\g++ -fPIC -c com\3lab\main.cpp -o com\3lab\main.o

C:\Programs>MinGW64\bin\g++ -o com\3lab\main.exe com\3lab\main.o -Lcom\3lab\. –lload

C:\Programs>C:\Programs\com\3lab\main.exe

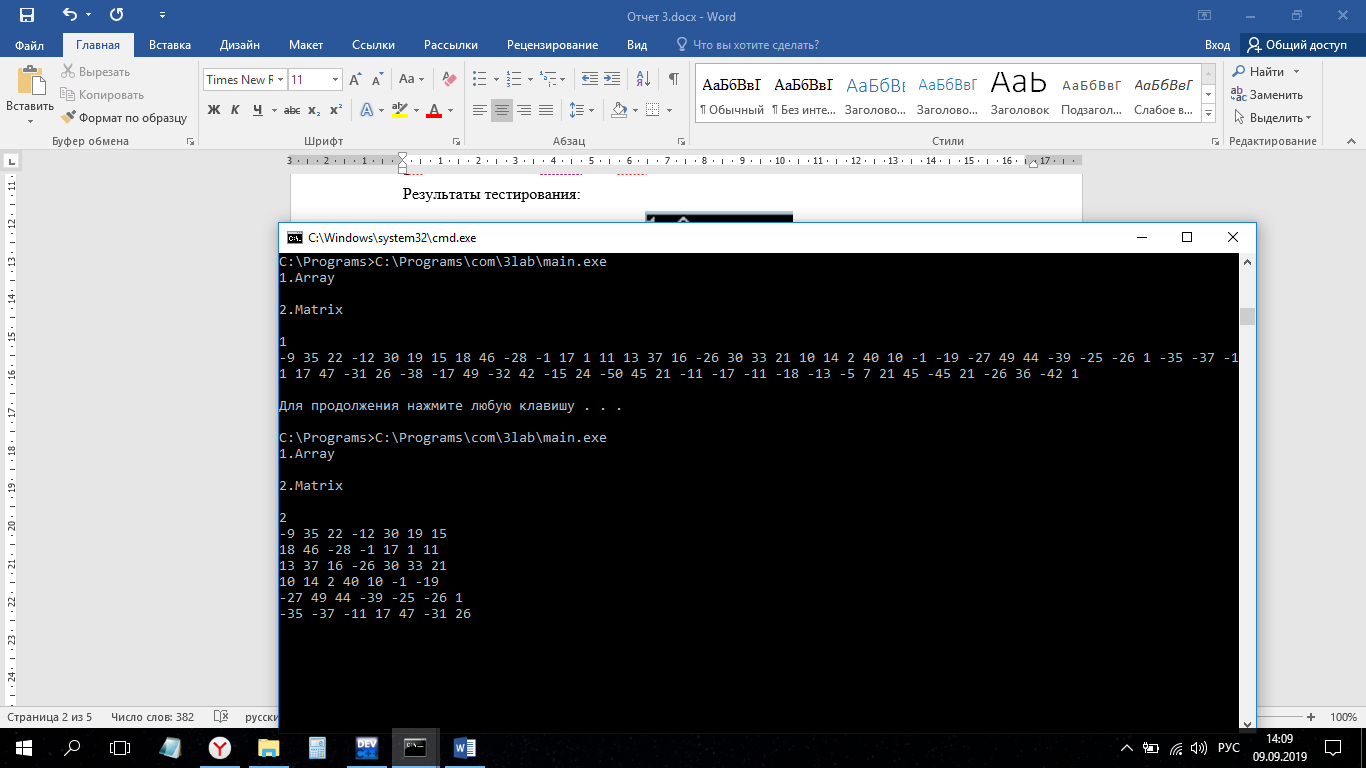
Результаты тестирования:

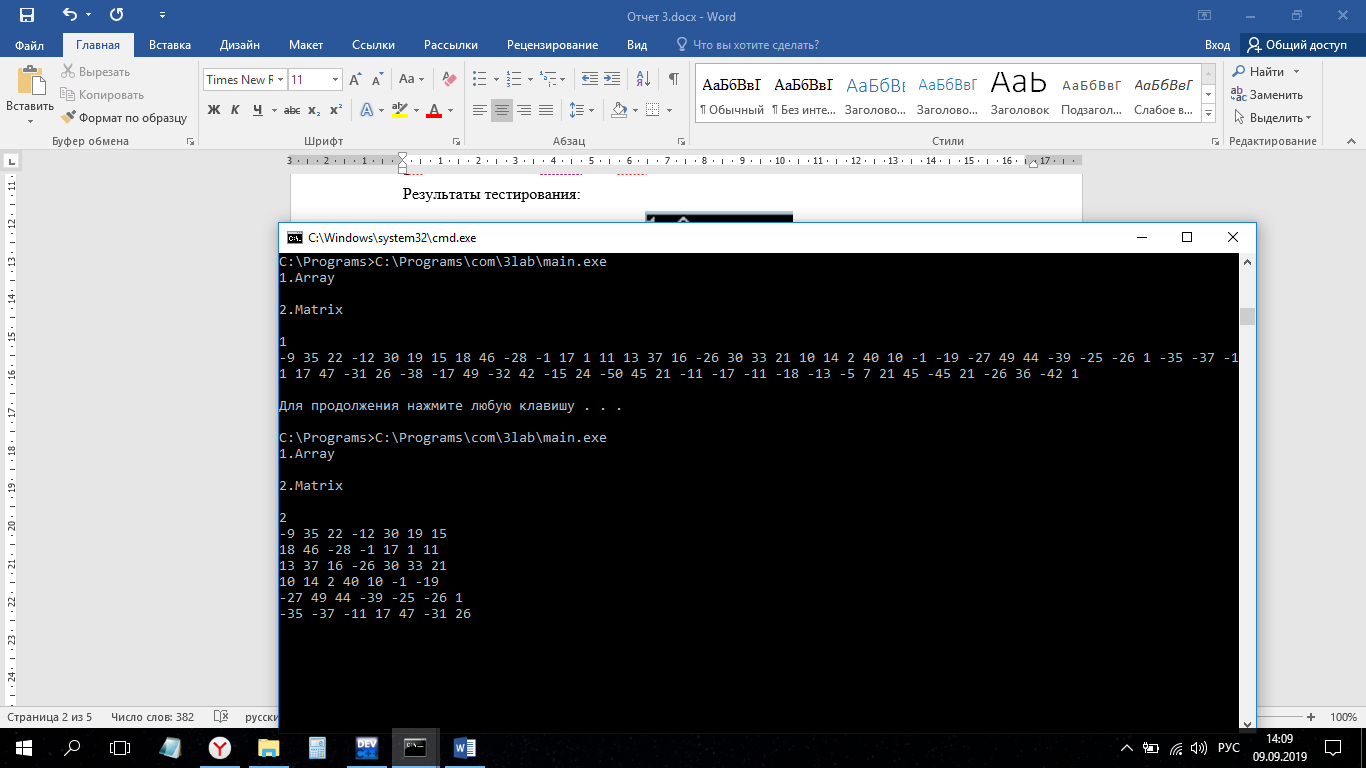
Рисунок 3.2 - Демонстрация меню выбора

Рисунок 3.3 - Демонстрация работы с библиотекой массива

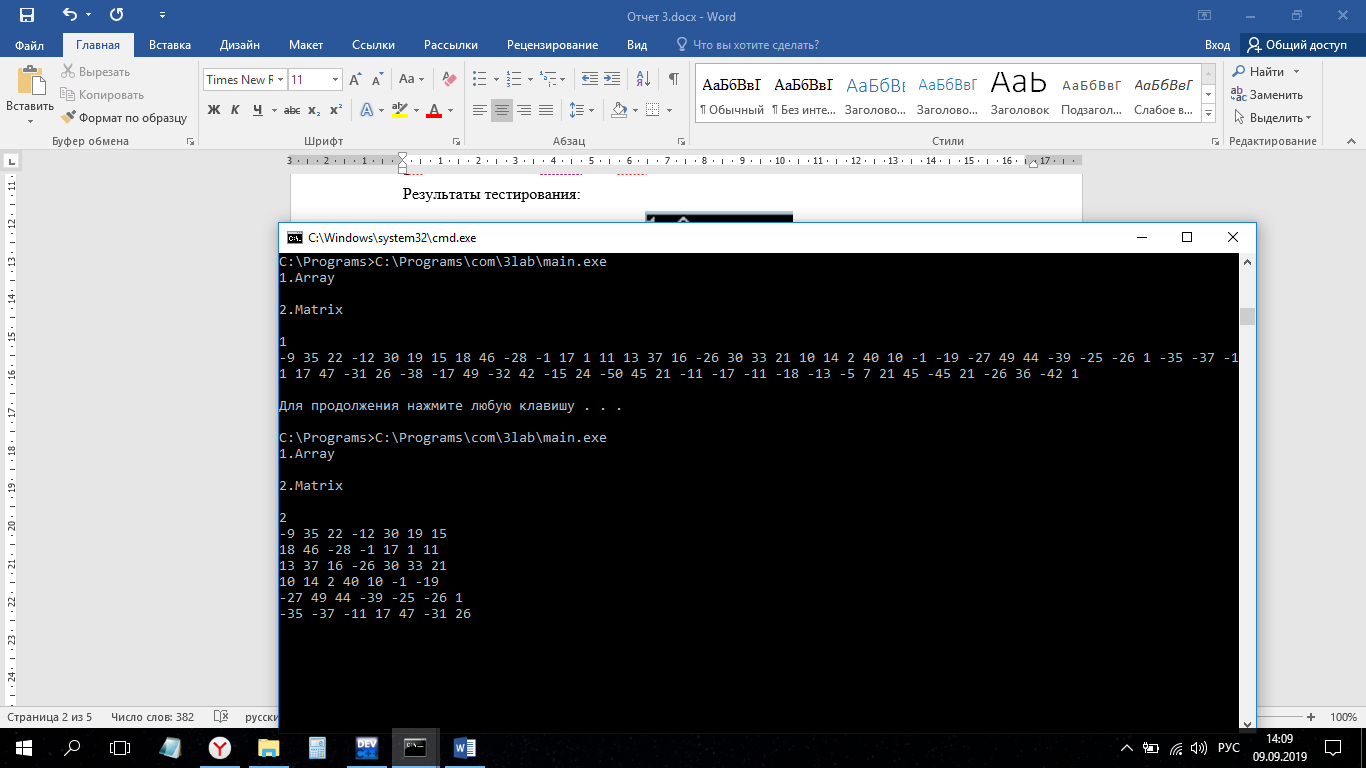


Рисунок 3.4 - Демонстрация работы с библиотекой матрицы

Текст программы:

*main.cpp*

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <math.h>

#include "load.h"

int main(void){

char num;

puts("1.Array\n");

puts("2.Matrix\n");

scanf("%c",&num);

if(num=='1'){

//load();

LoadRun("lib1.dll");

}else if(num=='2'){

LoadRun("lib2.dll");

//load();

}

system("pause");

return 0;

}

*load.cpp*

#include <stdio.h>

#include <stdlib.h>

#include "load.h"

#include "fun.h"

#ifndef WIN32

#include <dlfcn.h>

#else

#include <windows.h>

#endif

void LoadRun(const char \* const s)

{

void \*lib;

void (\*fun)(void);

lib = LoadLibrary(s);

if(!lib){

printf("cannot open library'%s'\n",s);

return;

}

#ifndef WIN32

fun = (void(\*)(void))dlsym(lib," "\_Z4loadv"");

#else

fun = (void(\*)(void))GetProcAddress((HINSTANCE)lib,"load");

#endif

if(fun==NULL)

{

printf("cannot load function\n");

}else

{

fun();

}

#ifndef WIN32

dlclose(lib);

#else

FreeLibrary((HINSTANCE)lib);

#endif

}

*matrix.cpp*

#include <cstring>

#include <cstdio>

#include <iostream>

#include <cstdlib>

#define N 6

#define M 7

#include "fun.h"

using namespace std;

void load()

{

int matrix[N][M];

for(int i = 0; i < N; i++)

{

for(int j = 0; j < M; j++)

{

matrix[i][j] = rand()%101-50;

}

}

printArr(matrix);

findArr(matrix);

}

void printArr(int matrix[N][M])

{

for(int i = 0; i < N; i++)

{

for(int j = 0; j < M; j++)

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

cout<<endl;

}

cout<<endl;

}

void findArr(int matrix[N][M])

{

for (int i = 0; i < N; i++)

{

for (int j = 0; j < M; j++)

if (matrix[i][j] < 0) matrix[i][j] \*= matrix[i][j];

cout<<endl;

}

cout<<endl;

}

*array.cpp*

#include <cstring>

#include <cstdio>

#include <iostream>

#include <cstdlib>

#define B 67

#include "fun.h"

using namespace std;

void load()

{

int array[B];

for(int i = 0; i < B; i++)

{

array[i] = rand()%101-50;

}

printArr(array);

findArr(array);

}

void printArr(int array[B])

{

for (int i = 0; i < B; i++)

cout<<array[i]<<" ";

cout<<endl;

}

void findArr(int array[B])

{

for (int i = 0; i < B; i++)

{

if (array[i] < 0) array[i] \*= array[i];

}

cout<<endl;

}