**ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ**

Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**МОСКОВСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**СВЯЗИ И ИНФОРМАТИКИ**

Кафедра информационной безопасности

**ЛАБОРАТОРНАЯ РАБОТА**

**№ 1**

по дисциплине «ООПСЗИ»

**«Компиляция файлов исходного кода и компоновка полученных объектных файлов в исполняемый модуль. Линейные алгоритмы»**

Выполнил студент группы БПЗ1801

Игумнов В.О.

Проверила

Барков В. В.

Москва, 2021

**Цель работы:**

Овладеть навыками создания однофайловых и многофайловых проектов в интегрированной среде разработки Microsoft Visual Studio 2015 Community Edition и научится создавать линейные программы на языке C с применением арифметических операций.

**Задание:**

По номеру Вашего варианта выбрать задачу, решаемую в этой лабораторной работе, и выполнить для нее следующие задания.

Вариант индивидуального задания:



**Выполнение:**

***Задание 1.***

Составить программу, имеющую линейный алгоритм и состоящую из одной функции **void main()**. Программу записать в task1.c. Скомпилировать, скомпоновать и выполнить.

В функции main организовать вычисление задачи вашего варианта дважды:

* Для исходных данных, значения которых задать в виде констант в тексте функции main.
* Для исходных произвольных данных, значения которых пользователь Вашей программы должен ввести с клавиатуры в процессе выполнения программы.

Вывод действительных чисел осуществлять с точностью до 0.0001.

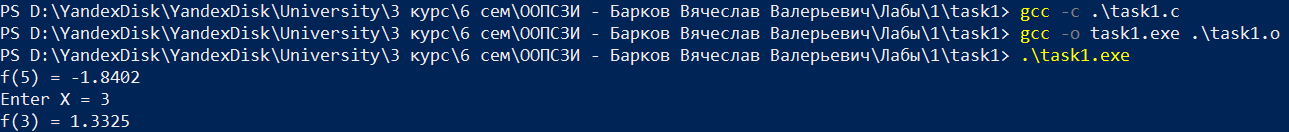
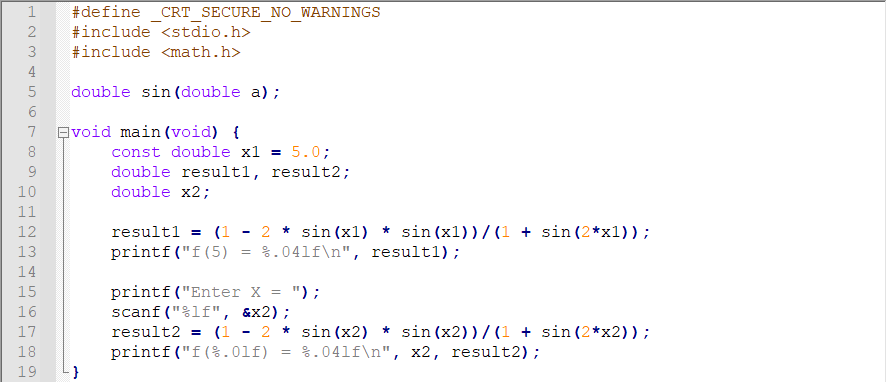


Рисунок 1 – Задание 1.

Листинг 1 – Исходный код файла task1.c

****

***Задание 2.***

Линейный алгоритм функции из задания 1 разделить на две процедуры, выделив в одну вычислительные операции этого алгоритма, а в другую все операции ввода-вывода. Каждую процедуру оформить как функцию. Вычислительную часть алгоритма оформить как функцию с параметрами, передаваемыми по значению, и возвращаемым значением. Прототип функции:

**double f(double x)** // Если функция имеет один параметр

**double f(double x, double y)** // Если функция имеет два параметра

Другую часть алгоритма оформить как функцию **void main()**, вызывающую первую функцию нужное количество раз. Записать тексты функций файл с именем task2.с в следующем порядке: функция с параметрами, функция main. Cкомпилировать, скомпоновать и выполнить.

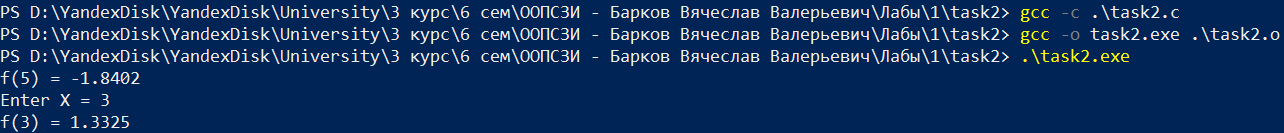
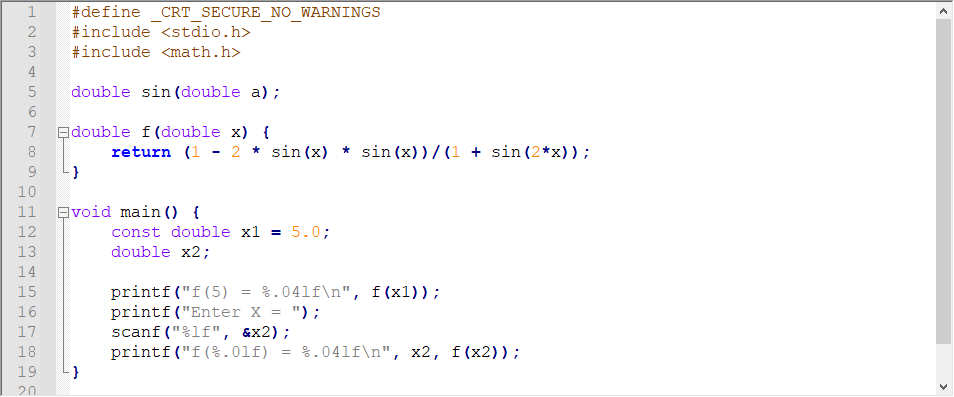


Рисунок 2 – Задание 2.

Листинг 2 – Исходный код файла task2.c

****

***Задание 3.***

Создать файл task3.с, в котором изменить порядок записи текстов функций, созданных в задании 2. Функции записать в следующем порядке: функция main, функция с параметрами, организующая вычисления **(double f(double x)** или **double f(double x, double y)).** Внести требуемые дополнения, добиться успешной компиляции, скомпоновать и выполнить.

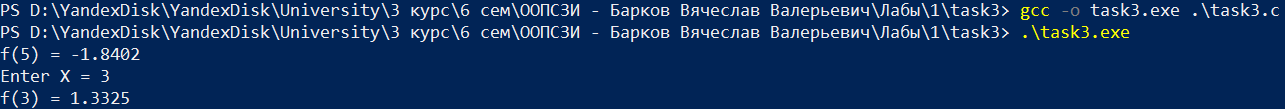
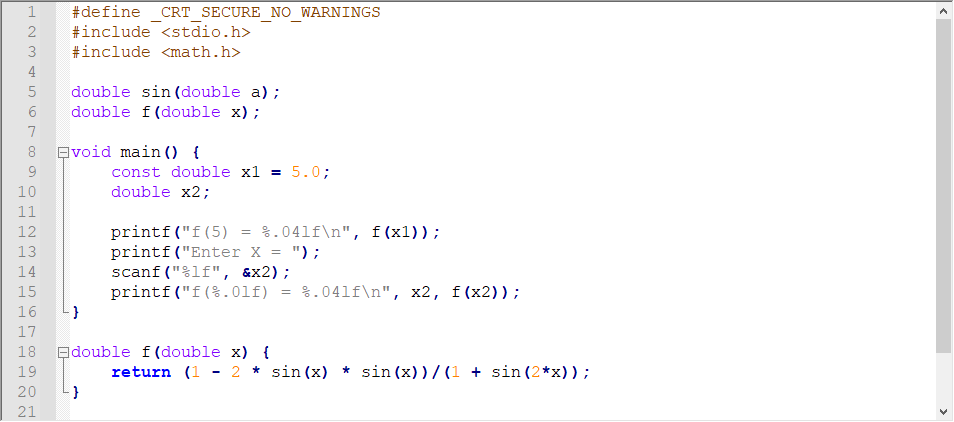
****

Рисунок 3 – Задание 3.

Листинг 3 – Исходный код файла task3.c

****

***Задание 4.***

По-прежнему линейный алгоритм функции из задания 1 разделить на две процедуры, выделив в одну вычислительные операции этого алгоритма, а в другую все операции ввода-вывода.

Вычислительную часть алгоритма оформить как функцию без параметров и без возвращаемого значения. Прототип функции **void f(void)**

Другую часть алгоритма оформить как функцию **void main()**, вызывающую вычислительную функцию. Обмен данными между функциями организовать через глобальные объекты (**double x, y** – аргументы, **double result** – результат вычисления). Записать тексты функций в файл с именем task4.с в следующем порядке: функция main, затем функция без возвращаемого значения и без параметров. Cкомпилировать, скомпоновать и выполнить.

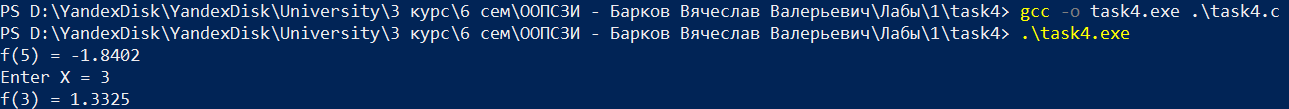
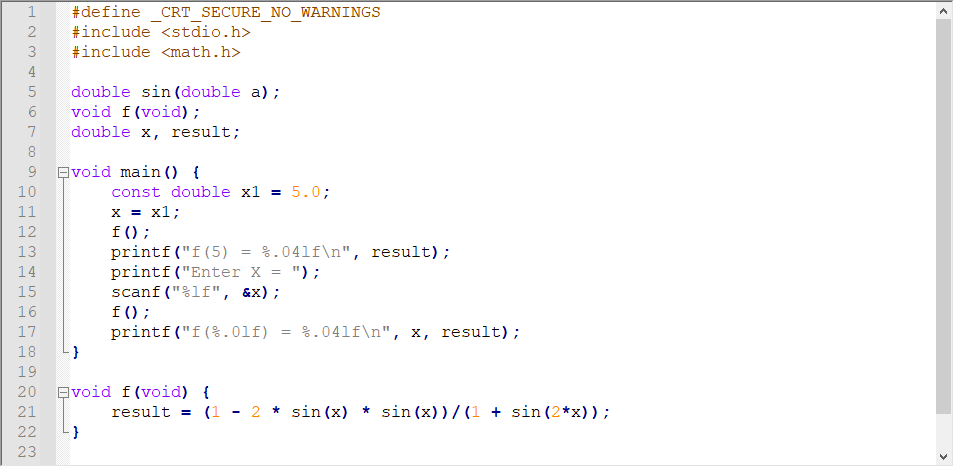
****

Рисунок 4 – Задание 4.

Листинг 4 – Исходный код файла task4.c

****

***Задание 5.***

В этом задании необходимо разделить текст файла task3.с на два файла. В первый файл с именем task5\_main.c поместить текст функции main. Скомпилировать только файл task5\_main.c.

Во второй файл с именем task5\_func.с поместить текст функции с параметрами (double f(double x) или double f(double x, double y)). Скомпилировать только файл task5\_func.c. После раздельной компиляции осуществить совместную компоновку. Полученный исполняемый файл выполнить.

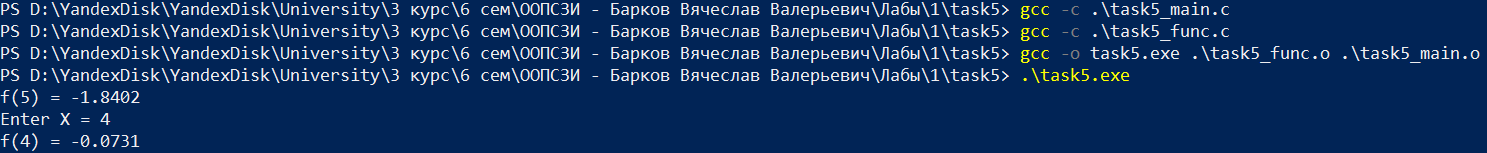
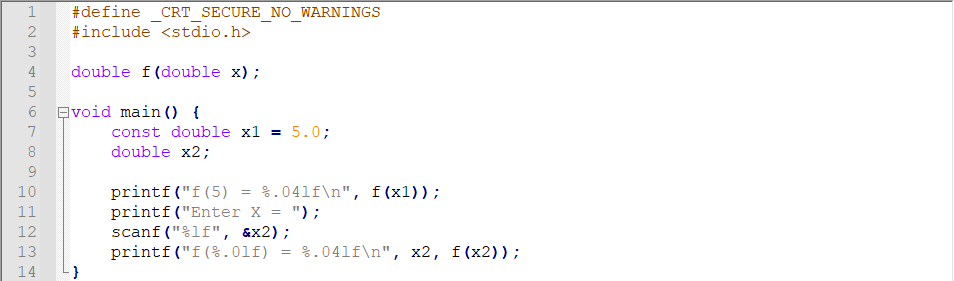
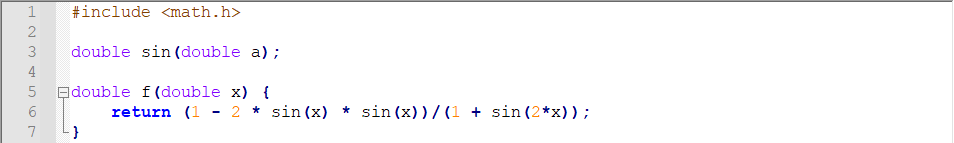
****

Рисунок 5 – Задание 5.

Листинг 5 – Исходный код файла task5\_main.c

****

Листинг 6 – Исходный код файла task5\_func.c

****

***Задание 6.***

В этом задании необходимо разделить текст файла task4.с на два файла.

В первый файл с именем task6\_main.c поместить текст функции main. Скомпилировать только файл task6\_main.c.

Во второй файл с именем task6\_func.с поместить определения глобальных объектов (double x, y – аргументы, double result – результат вычисления) и текст функции без параметров (void f(void)). Скомпилировать только файл task6\_func.c.

После раздельной компиляции осуществить совместную компоновку. Разобраться в проблемах, возникающих при совместной компиляции и компоновки. Полученный исполняемый файл выполнить.

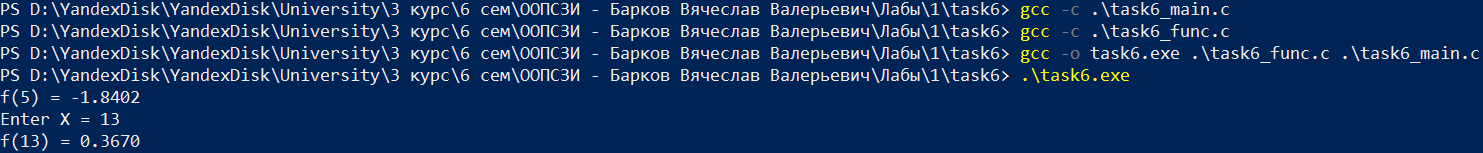
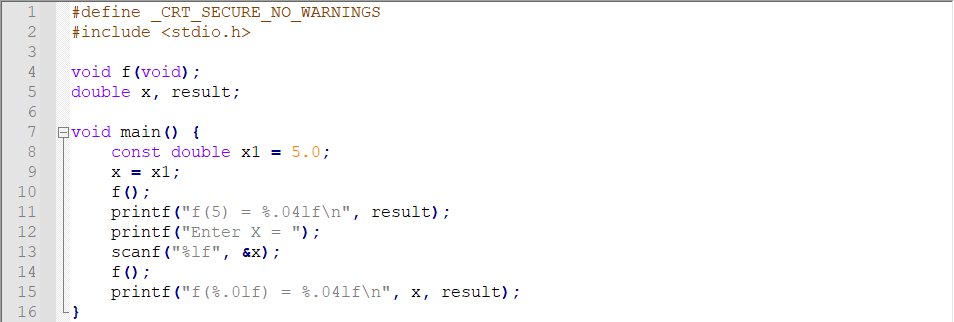
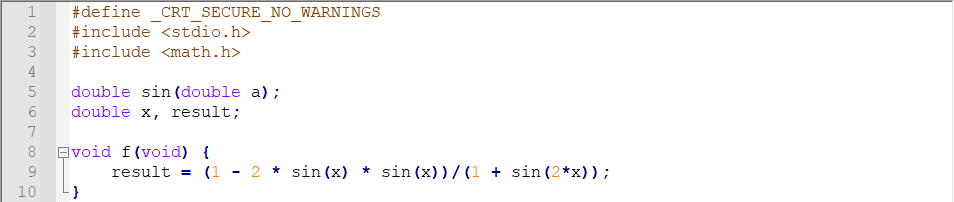
****

Рисунок 6 – Задание 6.

Листинг 7 – Исходный код файла task6\_main.c

****

Листинг 8 – Исходный код файла task6\_func.c

****

***Задание 7.***

В этом задании необходимо модифицировать тексты файлов из задания 6.

Описание функции и внешних переменных выделить в отдельный заголовочный файл func.h, включить его содержимое в файлы task7\_main.c и task7\_func.c. Определение внешних переменных произвести в файле task7\_func.c.

Скомпилировать по отдельности файлы task7\_main.c и task7\_func.c. Произвести компоновку, выполнить полученный исполняемый файл.

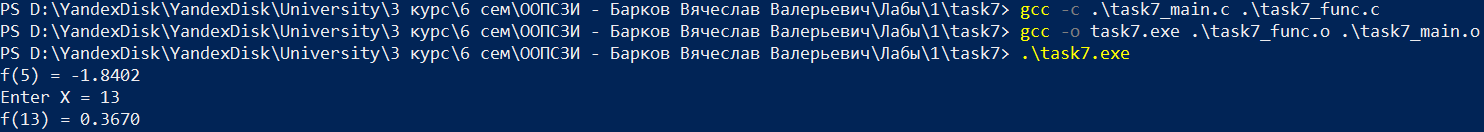
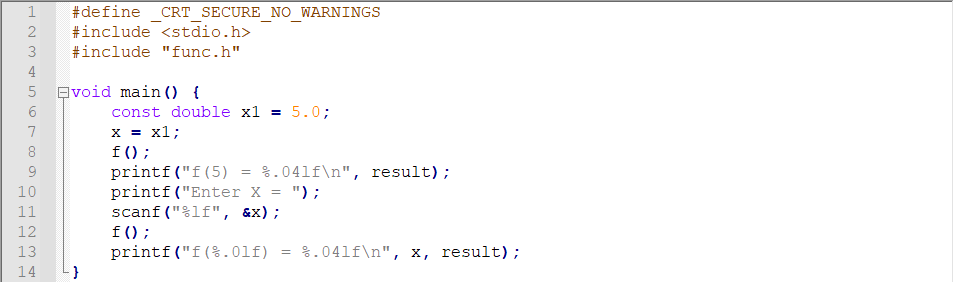
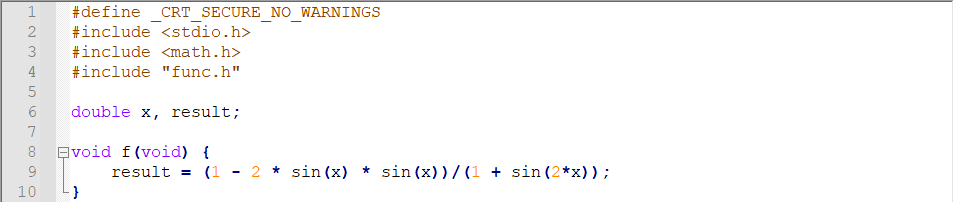


Рисунок 7 – Задание 7.

Листинг 9 – Исходный код файла task7\_main.c

****

Листинг 10 – Исходный код файла task7\_func.c

****

Листинг 11 – Исходный код файла func.h

****

***Задание 8.***

Скомпилировать по отдельности файлы из задания 7 task7\_main.c и task7\_func.c. Создать статическую библиотеку task8\_lib.lib, включающую объектный файл task7\_func.obj. Полученную статическую библиотеку скомпоновать с файлом task7\_main.obj. Выполнить полученный исполняемый файл.

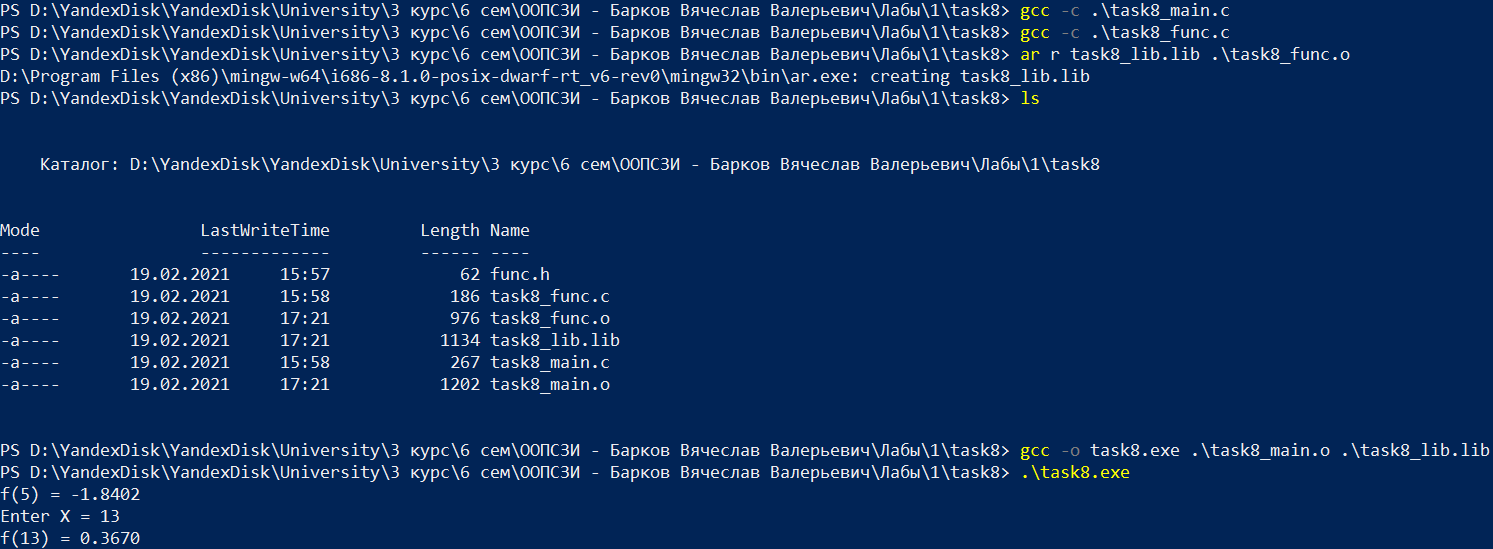
****

Рисунок 8 – Задание 8.

Ссылка на Github репозиторий: <https://github.com/Cno6/Igumnov_OOP_labs>