**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 «Объектно-ориентированное программирование система защиты информации»

на тему:

**«Процедурное программирование на языке программирования C»**

Вариант №16

Выполнил: студ. гр. БПЗ1501

Романюк И. Ф.

Проверил: ассисент кафедры ИБ

Барков В. В.

Москва 2018

**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.

**Выполнение**

Задание 1

Составить программу, имеющую линейный алгоритм и состоящую из одной функции int main(). Программу записать в файл с именем 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.

Исходный код программы:

#define \_CRT\_SECURE\_NO\_WARNINGS

#include "stdio.h"

#include "math.h"

void main()

{

double x = 10, f = (pow(x,2) + 2 \* x - 3 + (x + 1)\*sqrt(pow(x,2) - 9)) / (pow(x,2) - 2 \* x - 3 + (x - 1)\*sqrt(pow(x,2) - 9));

printf("x = %.4f\n f = %.4f\n\n", x, f);

printf("x = ");

scanf("%lf", &x);

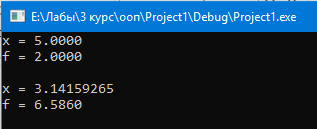
f = (pow(x, 2) + 2 \* x - 3 + (x + 1)\*sqrt(pow(x, 2) - 9)) / (pow(x, 2) - 2 \* x - 3 + (x - 1)\*sqrt(pow(x, 2) - 9));

printf("f = %.4f", f);

getch();

}

Результаты:



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 int main(), вызывающую первую функцию нужное количество раз. Записать тексты функций в файл с именем task2.с в следующем порядке: функция с параметрами, функция main. Cкомпилировать, скомпоновать и выполнить.

Исходный код программы:

#define \_CRT\_SECURE\_NO\_WARNINGS

#include "stdio.h"

#include "math.h"

double f(double x)

{

return (pow(x, 2) + 2 \* x - 3 + (x + 1)\*sqrt(pow(x, 2) - 9)) / (pow(x, 2) - 2 \* x - 3 + (x - 1)\*sqrt(pow(x, 2) - 9));

}

void main()

{

double x = 5;

printf("x = %.4f\nf = %.4f\n\n", x, f(x));

printf("x = ");

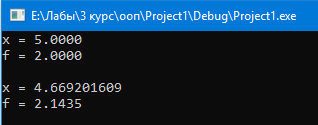
scanf("%lf", &x);

printf("f = %.4f", f(x));

getch();

}

Результаты:



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.

Исходный код программы:

#define \_CRT\_SECURE\_NO\_WARNINGS

#include "stdio.h"

#include "math.h"

double f(double x);

void main()

{

double x = 5;

printf("x = %.4f\nf = %.4f\n\n", x, f(x));

printf("x = ");

scanf("%lf", &x);

printf("f = %.4f", f(x));

getch();

}

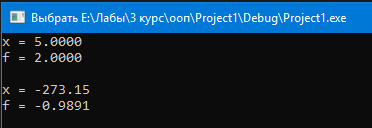
double f(double x)

{

return (pow(x, 2) + 2 \* x - 3 + (x + 1)\*sqrt(pow(x, 2) - 9)) / (pow(x, 2) - 2 \* x - 3 + (x - 1)\*sqrt(pow(x, 2) - 9));

}

Выполнение:



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 int 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.

Исходный код:

#define \_CRT\_SECURE\_NO\_WARNINGS

#include "stdio.h"

#include "math.h"

double x, result;

void f();

void main()

{

x = 5;

f();

printf("x = %.4f\nf = %.4f\n\n", x, result);

printf("x = ");

scanf("%lf", &x);

f();

printf("f = %.4f", result);

getch();

}

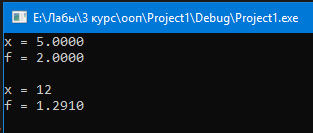
void f()

{

result=(pow(x, 2) + 2 \* x - 3 + (x + 1)\*sqrt(pow(x, 2) - 9)) / (pow(x, 2) - 2 \* x - 3 + (x - 1)\*sqrt(pow(x, 2) - 9));

}

Результат:



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.

Проанализировать результаты работы и сделать выводы.

Исходный код программы:

Файл task5\_func.c:

#include "math.h"

double f(double x)

{

return (pow(x, 2) + 2 \* x - 3 + (x + 1)\*sqrt(pow(x, 2) - 9)) / (pow(x, 2) - 2 \* x - 3 + (x - 1)\*sqrt(pow(x, 2) - 9));

}

Файл task5\_main.c:

#define \_CRT\_SECURE\_NO\_WARNINGS

#include "stdio.h"

#include "math.h"

double f(double x);

void main()

{

double x = 5;

printf("x = %.4f\nf = %.4f\n\n", x, f(x));

printf("x = ");

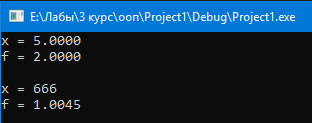
scanf("%lf", &x);

printf("f = %.4f", f(x));

getch();

}

Результат:



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. Проанализировать результаты работы и сделать выводы.

Исходный код программы:

Файл task5\_func.c:

#include "math.h"

double x, result;

void f()

{

result = (pow(x, 2) + 2 \* x - 3 + (x + 1)\*sqrt(pow(x, 2) - 9)) / (pow(x, 2) - 2 \* x - 3 + (x - 1)\*sqrt(pow(x, 2) - 9));

}

Файл task5\_main.c:

#define \_CRT\_SECURE\_NO\_WARNINGS

#include "stdio.h"

#include "math.h"

void f();

extern double x, result;

void main()

{

x = 5;

f();

printf("x = %.4f\nf = %.4f\n\n", x, result);

printf("x = ");

scanf("%lf", &x);

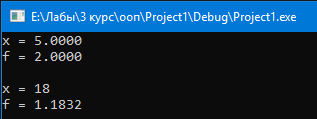
f();

printf("f = %.4f", result);

getch();

}

Результат:



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.

Исходный код программы:

Файл func.h:

#pragma once

#define \_CRT\_SECURE\_NO\_WARNINGS

#include "stdio.h"

#include "math.h"

extern double x, result;

void f();

Файл task7\_func.c:

#include "func.h"

double x, result;

void f()

{

result = (pow(x, 2) + 2 \* x - 3 + (x + 1)\*sqrt(pow(x, 2) - 9)) / (pow(x, 2) - 2 \* x - 3 + (x - 1)\*sqrt(pow(x, 2) - 9));

}

Файл task7\_main.c:

#include "func.h"

void main()

{

x = 5;

f();

printf("x = %.4f\nf = %.4f\n\n", x, result);

printf("x = ");

scanf("%lf", &x);

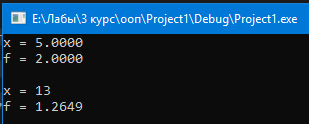
f();

printf("f = %.4f", result);

getch();

}

Результат:



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.