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**»**

Выполнила студентка

Группы БПЗ 1501

Рамазанова Эвелина

Москва, 2018

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

**Задание:**

Выбрав вариант задачи, решаемой в данной лабораторной работе, выполним для нее, перечисленные ниже задания.

2. 𝑓(𝑥) = cos(𝑥) + sin(𝑥) + cos(3𝑥) + sin(3𝑥)

**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 файла task1.c

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <math.h>

int main(void)

{

double x, func;

x = 3.14/2;

func = cos(x) + sin(x) + cos(3 \* x) + sin(3 \* x);

printf("x=pi/2 Answer is = %.4lf\n", func);

printf("Enter x=");

scanf("%lf", &x);

func = cos(x) + sin(x) + cos(3 \* x) + sin(3 \* x);

printf("Answer is = %.4lf", func);

getchar();

getchar();

return(0);

}



Рис. 1 Окно программы

**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.

Исходный код файла task2.c

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <math.h>

double func(double x)

{

double func;

func = cos(x) + sin(x) + cos(3\*x) + sin(3\*x);

return(func);

}

int main(void)

{

double x, y;

x = 3.14/2;

y = func(x);

printf("x=3.14/2 Answer = %.4lf\n", y);

printf("Enter x=");

scanf("%lf", &x);

y = func(x);

printf("Answer = %.4lf", y);

getchar();

getchar();

return(0);

}



Рис. 2 Окно реализации программы

**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 файла task3.c

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <math.h>

double func(double x);

int main(void)

{

double x, y;

x = 3.14;

y = func(x);

printf("x=3.14 Answer = %.4lf\n", y);

printf("Enter x=");

scanf("%lf", &x);

y = func(x);

printf("Answer = %.4lf", y);

getchar();

getchar();

return(0);

}

double func(double x)

{

double func;

func = cos(x) + sin(x) + cos(3\*x) + sin(3\*x);

return(func);

}



Рис. 3

**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 файла task4.c

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <math.h>

double func();

double x, a;

int main(void)

{

x = 3.14/2;

func();

printf("x=3.14/2 Answer = %.4lf\n", a);

printf("Enter x=");

scanf("%lf", &x);

func();

printf("Answer = %.4lf", a);

getchar();

getchar();

return(0);

}

double func(void)

{

a = cos(x)+ sin(x)+ cos(3\*x)+ sin(3\*x);

return(0);

}



Рис. 4

**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.

Исходный код файла task5-main.c

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <math.h>

double func(double x);

int main(void)

{

double x, a;

x = 3.14/2;

a = func(x);

printf("x=3.14/2 Answer = %.4lf\n", a);

printf("Enter x=");

scanf("%lf", &x);

a = func(x);

printf("Answer = %.4lf", a);

getchar();

getchar();

return(0);

}

Исходный код файла task5-func.с

#include <stdio.h>

#include <math.h>

double func(double x)

{

double y;

y = cos(x) + sin(x) + cos(3 \* x) + sin(3 \* x);

return(y);

}



Рис. 5

**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.

Исходный код файла task6-main.c

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <math.h>

double x, a;

void func(void);

int main(void)

{

x = 3.14/2;

func();

printf("x=3.14/2 Answer = %.4lf\n", a);

printf("Enter x=");

scanf("%lf", &x);

func();

printf("Answer = %.4lf", a);

getchar();

getchar();

return(0);

}

Исходный код файла task6-func.c

#include <math.h>

double a, x;

double func(void)

{

a = cos(x) + sin(x) + cos(3\*x) + sin(3\*x);

return(0);

}



Рис. 6

**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.

Исходный код файла task7-main.c

#define \_CRT\_SECURE\_NO\_WARNINGS

#include "func.h"

#include <stdio.h>

#include <math.h>

int main(void)

{

x = 3.14/2;

func();

printf("x=3.14/2 Answer is = %.4lf\n", a);

printf("Enter x=");

scanf("%lf", &x);

func();

printf("Answer is = %.4lf", a);

getchar();

getchar();

return(0);

}

Исходный код файла task7-func.c

#include "func.h"

#include <math.h>

double func(void)

{

a = cos(x) + sin(x) + cos(3\*x) + sin(3\*x);

}

Исходный код файла func.h

#pragma once

double x, a;

double func();



Рис. 7

**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.

