Министерство цифрового развития, связи и массовых коммуникаций Российской Федерации

Ордена Трудового Красного Знамени федеральное государственное

бюджетное образовательное учреждение

высшего образования

«Московский технический университет связи и информатики»

Кафедра «Информационная безопасность»

Лабораторная №1

Выполнила:

студентка группы БП31801

Зелик М.Б.

Вариант №10

Москва, 2021

**1. Общее задание**

**Задание**

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

***Задание 1***

Составить программу, имеющую линейный алгоритм и состоящую из одной функции void 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.

Проанализировать результаты работы и сделать выводы. Перенести полученные результаты в отчет.

Вывод действительных чисел осуществлять с точностью до 0.0001

Примеры вывода:

|  |  |
| --- | --- |
| Для функции одной переменной | Для функции двух переменных |
| x = 5  y = 8.2537  f = 0.4514  x = 3.0051  y = 21  f = 84.5201 | x = 5  f = 0.4514  x = 3.0051  f = 84.5201 |

***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.

***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.

***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.

***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.

***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.

***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.

***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.

**2. Индивидуальное задание**

Вариант №10

**3. Выполнение**

**Задание 1**

#include "stdio.h"

#include "stdlib.h"

#define \_USE\_MATH\_DEFINES

#include <math.h>

void main()

{

double x = 5;

double f;

f = (sin((M\_PI/2)+3\*x))/(1-sin(3\*x-M\_PI));

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

printf ("f = %5.4f\n",f);

x = 0;

printf("Input x1: ");

scanf("%lf",&x);

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

f = (sin((M\_PI/2)+3\*x))/(1-sin(3\*x-M\_PI));

printf("f1 = %5.4f\n",f);

system("pause");

return;

}

Листинг 1. Программный код task1.c

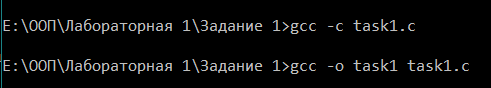


Рис 1. Сборка task1

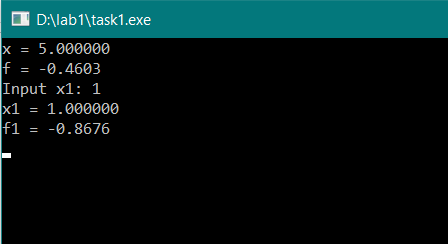


Рис 2. Результат выполнения task1

**Задание 2**

#include "stdio.h"

#include "stdlib.h"

#define \_USE\_MATH\_DEFINES

#include <math.h>

double f(double x)

{

return (sin((M\_PI/2)+3\*x)/(1-sin(3\*x-M\_PI)));

}

void main ()

{

double x = 5;

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

printf ("f = %5.4f\n",f(x));

x = 0;

printf("Input x1: ");

scanf("%lf",&x);

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

printf("f1 = %5.4f\n",f(x));

system("pause");

return;

}

Листинг 2. Программный код task2.c

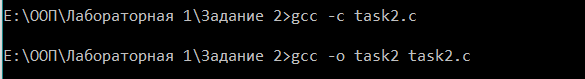


Рис 3. Сборка task2

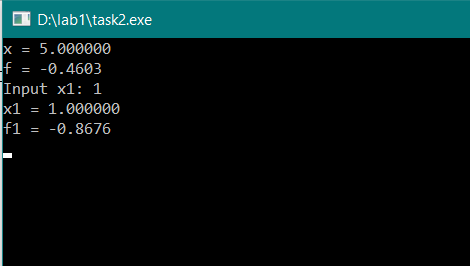


Рис 4. Результат выполнения task2

**Задание 3**

#include "stdio.h"

#include "stdlib.h"

#define \_USE\_MATH\_DEFINES

#include <math.h>

double f(double x);

void main (void)

{

double x = 5;

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

printf ("f = %5.4f\n",f(x));

printf("Input x: ");

scanf("%lf",&x);

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

printf("f = %5.4f\n",f(x));

system("pause");

return;

}

double f(double x)

{

return (sin((M\_PI/2)+3\*x))/(1-sin(3\*x-M\_PI));

}

Листинг 3. Программный код task3.c

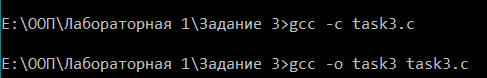


Рис 5. Сборка task3

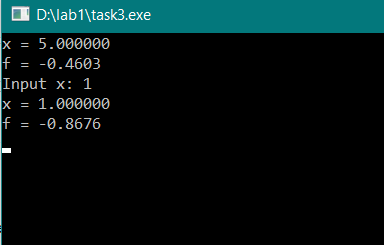


Рис 6. Результат выполнения task3

**Задание 4**

#include "stdio.h"

#include "stdlib.h"

#define \_USE\_MATH\_DEFINES

#include <math.h>

void f(void);

double x = 5;

double result;

void main ()

{

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

f();

printf ("f = %5.4f\n",result);

printf("Input x: ");

scanf("%lf",&x);

f();

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

printf("f = %5.4f\n",result);

system("pause");

}

void f(void)

{

result = (sin((M\_PI/2)+3\*x))/(1-sin(3\*x-M\_PI));

}

Листинг 4. Программный код task4.c

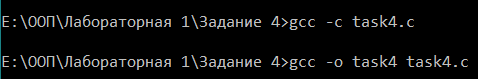


Рис 7. Сборка task4

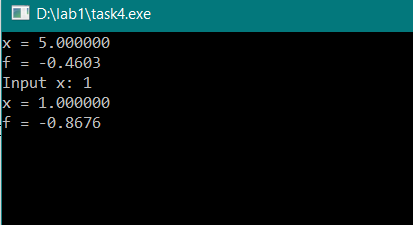


Рис 8. Результат выполнения task4

**Задание 5**

Программный код

#include "stdio.h"

#include "stdlib.h"

extern double f(double x);

void main (void)

{

double x = 5;

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

printf ("f = %5.4f\n",f(x));

printf("Input x: ");

scanf("%lf",&x);

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

printf("f = %5.4f\n",f(x));

system("pause");

return;

}

Листинг 5. Программный код task5\_main.c

#define \_USE\_MATH\_DEFINES

#include <math.h>

double f(double x)

{

return (sin((M\_PI/2)+3\*x))/(1-sin(3\*x-M\_PI));

}

Листинг 6. Программный код task5\_func.c

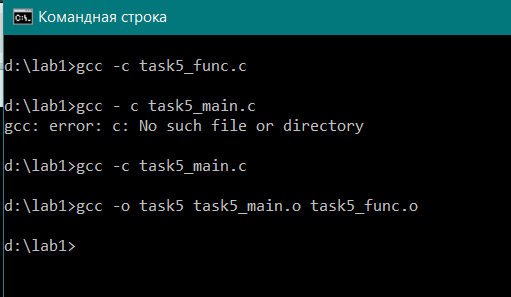


Рис 9. Сборка task5

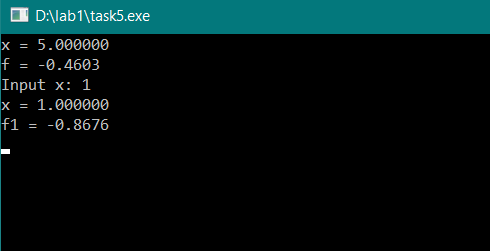


Рис 10. Результат выполнения task5

**Задание 6**

Программный код

#include "stdio.h"

#include "stdlib.h"

#define \_USE\_MATH\_DEFINES

#include <math.h>

void f (void);

double x;

double result;

void main()

{

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

f();

printf ("f=%5.4f\n",result);

printf ("Input x: ");

scanf ("%lf",&x);

f();

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

printf (" f=%5.4f\n",result);

system("pause");

}

Листинг 7. Программный код task6\_main.c

#include "stdio.h"

#define \_USE\_MATH\_DEFINES

#include <math.h>

double x = 5.0;

double result;

void f (void)

{

result = (sin((M\_PI/2)+3\*x))/(1-sin(3\*x-M\_PI));

}

Листинг 8. Программный код task6\_func.c

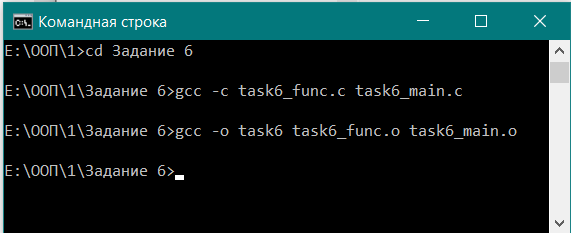


Рис 11. Сборка task6

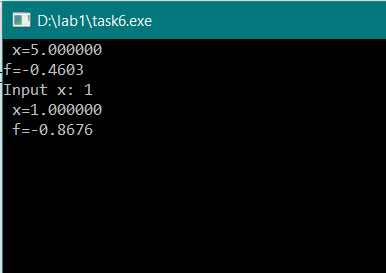


Рис 12. Результат выполнения task6

**Задание 7**

Программный код

#include "stdio.h"

#include "stdlib.h"

#define \_USE\_MATH\_DEFINES

#include <math.h>

#include "func.h"

void main()

{

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

f2();

printf ("f=%5.4f\n",result);

printf ("Input x: ");

scanf ("%lf",&x);

f2();

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

printf (" f=%5.4f\n",result);

system("pause");

}

Листинг 9. Программный код task7\_main.c

#include "stdio.h"

#define \_USE\_MATH\_DEFINES

#include <math.h>

#include "func.h"

double x = 5;

double result;

void f2 (void)

{

result = (sin((M\_PI/2)+3\*x))/(1-sin(3\*x-M\_PI));

}

Листинг 10. Программный код task7\_func.c

void f2 (void);

extern double x;

extern double result;

Листинг 11. Программный код func.h

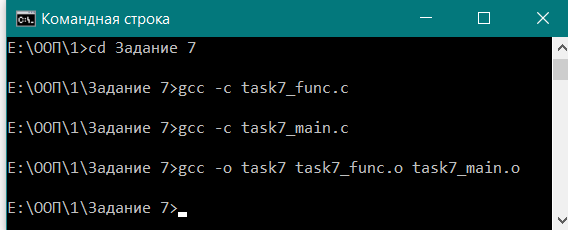


Рис 13. Сборка task7

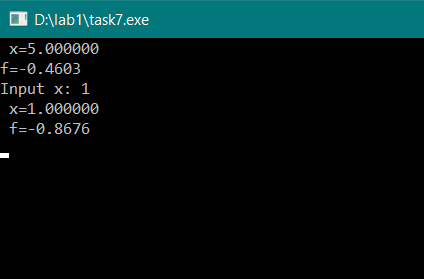


Рис 14. Результат выполнения task7

**Задание 8**

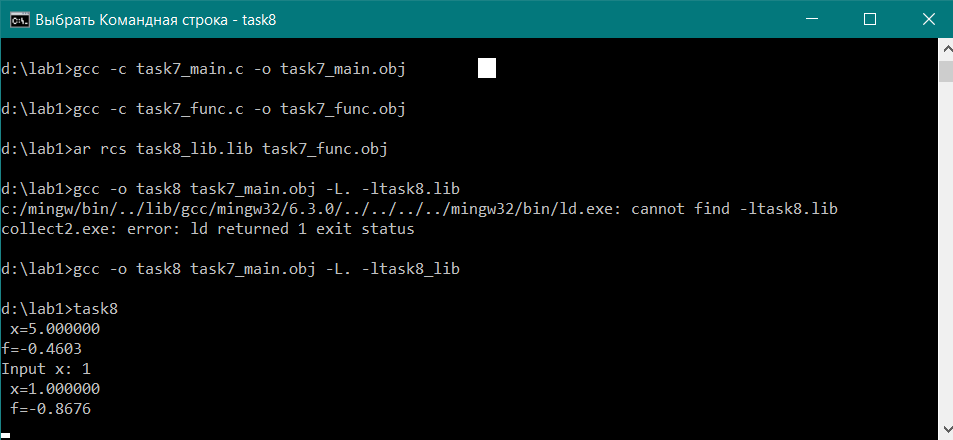


Рис 11. Сборка и результат выполнения task8