**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**

**№ 3**

по дисциплине «ООПСЗИ»

**«Регулярные и итерационные циклы»**

Выполнил

студент группы БПЗ1801

Игумнов В.О.

Проверил

Барков В. В.

Москва, 2021

**Цель работы:**

Овладеть навыками создания циклических алгоритмов на языке C с применением инструкций цикла на примере алгоритмов работы с последовательностями чисел.

**Задание:**

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

Все указанные ниже функции разработать в трёх вариантах: с использованием цикла for, while, do … while. Функциям из задания дать одинаковые имена. Каждую функцию поместить в отдельный файл. Прототипы функции вынести в заголовочный файл. В решении создать три проекта.

Индивидуальный вариант задания:

****

**Выполнение:**

***Задание 1.***

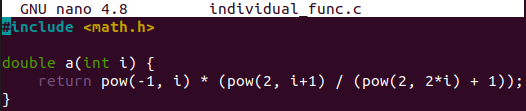
Написать функцию для вычисления выражения  Функция в качестве параметра принимает значение **n** и вычисляет сумму *n* слагаемых.

Прототип функции **double sum(int n).**

Функция не должна использовать функции консольного ввода-вывода.

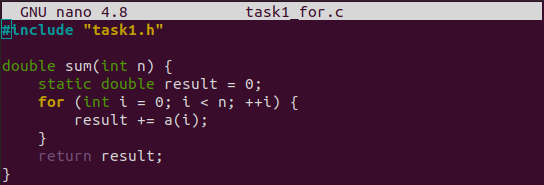
1. Создал отдельный файл для кода функции индивидуального задания с названием **individual\_func.c**.

Листинг 1 – Исходный код файла individual\_func.c

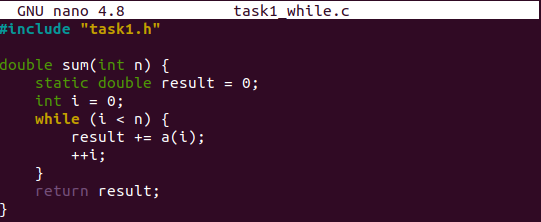


1. Написал функции для вычисления выражения в трёх вариантах с использованием **for**, **while**, **do … while**.

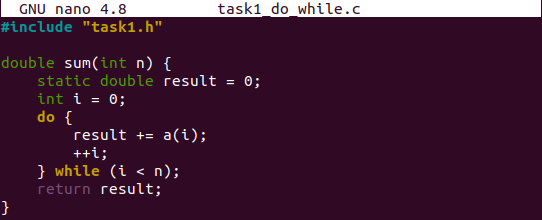
Листинг 2.1 – Исходный код файла task1\_for.c



Листинг 2.2 – Исходный код файла task1\_while.c



Листинг 2.3 – Исходный код файла task1\_do\_while.c



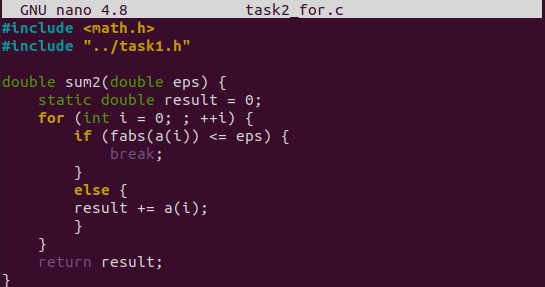
***Задание 2.***

Написать функцию для вычисления выражения с точность **ε**. Функция в качестве параметра принимает значение **ε** и вычисляет сумму до тех пор, пока не выполнится условие 

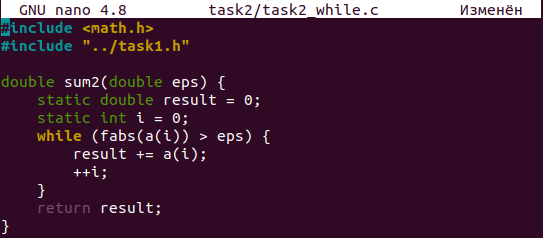
Прототип функции **double sum2(double eps).**

Функция не должна использовать функции консольного ввода-вывода.

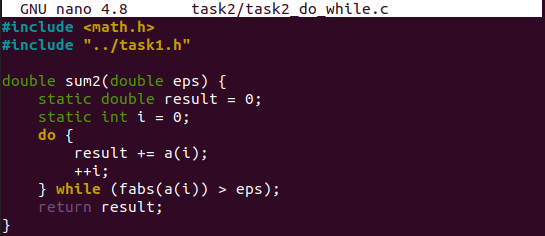
Листинг 3.1 – Исходный код файла task2\_for.c

****

Листинг 3.2 – Исходный код файла task2\_while.c

****

Листинг 3.1 – Исходный код файла task2\_do\_while.c

****

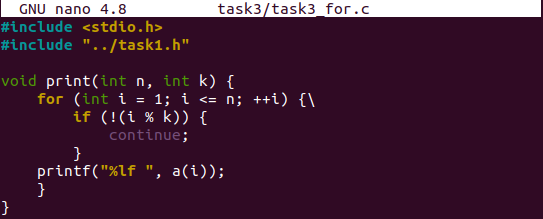
***Задание 3.***

Написать функцию, которая печатает **n** членов последовательности **{an}**, исключая из неё каждый **k**-ый член.

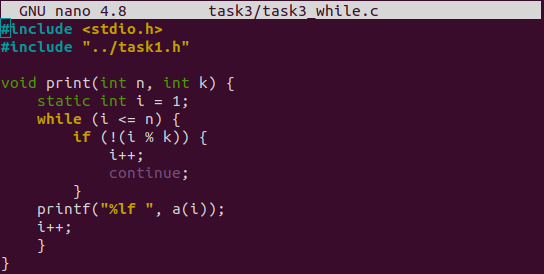
Числа **n** и **k** передаются в функцию в качестве параметра. Для исключения члена последовательности используйте инструкцию перехода **continue**.

Прототип функции **void print(int n, int k).**

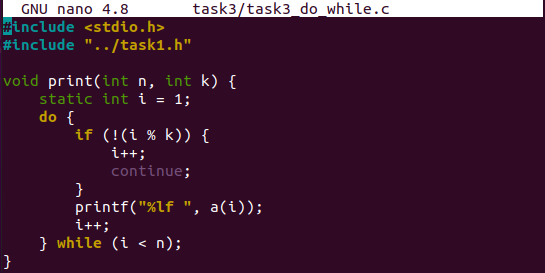
Листинг 4.1 – Исходный код файла task3\_for.c

****

Листинг 4.2 – Исходный код файла task3\_while.c

****

Листинг 4.3 – Исходный код файла task3\_while\_do.c

****

***Задание 4.***

Написать функцию, которая возвращает номер первого члена последовательности **{ai}**, для которого выполняется условие 

Функция в качестве параметра принимает значение **ε**. Выход из цикла организовать с помощью инструкции перехода **break**.

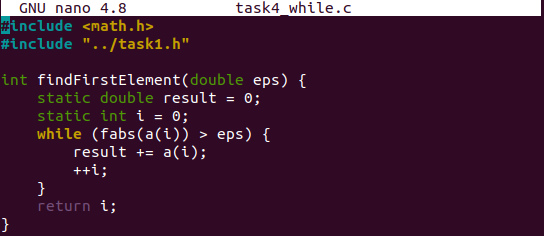
Прототип функции **int findFirstElement(double eps).**

Функция не должна использовать функции консольного ввода-вывода.

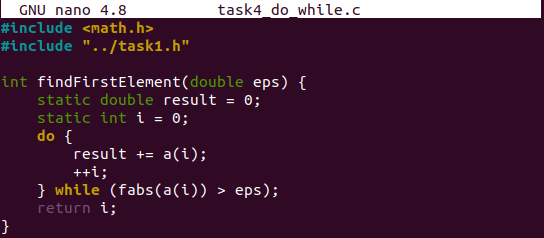
Листинг 5.1 – Исходный код файла task4\_for.c

****

Листинг 5.2 – Исходный код файла task4\_while.c

****

Листинг 5.3 – Исходный код файла task4\_do\_while.c

****

***Задание 5.***

Написать функцию, которая возвращает номер первого отрицательного члена последовательности **{ai­}**, для которого выполняется условие 

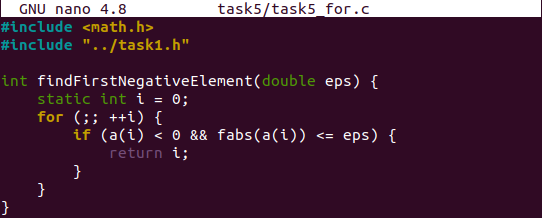
Функция в качестве параметра принимает значение **ε**.

Выход из цикла организовать с помощью инструкции перехода **return**.

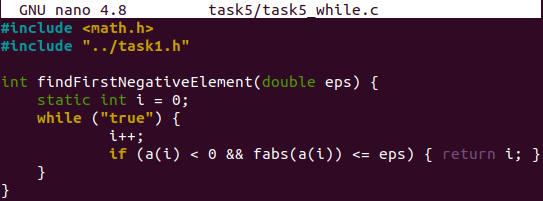
Прототип функции **int findFirstNegativeElement(double eps).**

Функция не должна испольовать функции консольного ввода-вывода.

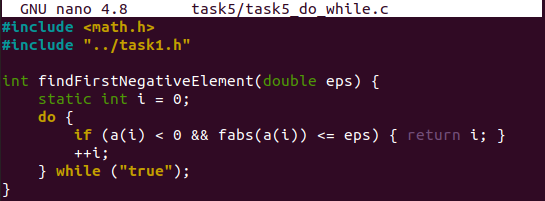
Листинг 6.1 – Исходный код файла task5\_for.c

****

Листинг 6.2 – Исходный код файла task5\_while.c

****

Листинг 6.3 – Исходный код файла task5\_do\_while.c

****

***Задание 6.***

Разработать функцию **main()**, которая в цикле выводит на экран меню из 6 пунктов и ожидает ответа пользователя:

1. Задание 1
2. Задание 2
3. Задание 3
4. Задание 4
5. Задание 5
6. Выход

Обработку пользовательского ввода выполнять с помощью инструкции **switch**. При нажатии на соответствующую цифру меню стирается с экрана и приложение переходит в режим демонстрации работы выбранного задания. По завершении демонстрации программа ожидает пользовательского ввода для перехода в меню. Выход из программы должно осуществляться при нажатии цифры 6 в режиме меню. Скомпилировать 3 версии программы:

1. Все функции реализованы с помощью цикла **for**.
2. Все функции реализованы с помощью цикла **while**.
3. Все функции реализованы с помощью цикла **do … while**

Убедиться, что все три версии программы работают одинаково.

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