ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ

**Федеральное государственное образовательное бюджетное учреждение**

**высшего профессионального образования**

**«Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича»**

Факультет Информационных технологий и программной инженерии

Кафедра Программной инженерии и вычислительной техники

КУРСОВАЯ РАБОТА

по дисциплине:

**«Программирование»**

тема: Анализ сигнала на выходе электрической цепи

Передаточная характеристика – 17 вариант

Входной сигнал – 5 вариант

Выполнил студент группы ИКПИ-41:

Панченков Е.Д. \_\_\_\_\_\_\_\_

Дата выполнения: «13» Май

Проверил:

Хазиев Н.Н. \_\_\_\_\_\_\_\_

Санкт-Петербург

2025

# Оглавление

[Оглавление 2](#_Toc198252440)

[Задание к курсовой работе 4](#_Toc198252441)

[Аннотация 5](#_Toc198252442)

[1) Общие сведения 7](#_Toc198252443)

[1.1) Обозначение и наименование программы 7](#_Toc198252444)

[1.2) Языки разработки: 7](#_Toc198252445)

[2) Функциональное назначение 7](#_Toc198252446)

[2.1) Решаемые задачи 7](#_Toc198252447)

[2.2) Назначение: 8](#_Toc198252448)

[2.3) Ограничения в функциональности: 8](#_Toc198252449)

[3) Контрольный расчёт 10](#_Toc198252450)

[4) Используемые технические средства 12](#_Toc198252451)

[5) Таблица идентификаторов 13](#_Toc198252452)

[6) Описание функций 14](#_Toc198252453)

[6.1) Таблица 6.1 14](#_Toc198252454)

[6.2) Таблица 6.2 14](#_Toc198252455)

[7) Алгоритм работы программы 16](#_Toc198252456)

[7.1) Блок схема 16](#_Toc198252457)

[7.2) Описание логической структуры программы 16](#_Toc198252458)

[a) Инициализация и запуск 16](#_Toc198252459)

[b) Обработка пользовательского ввода 17](#_Toc198252460)

[c) Выполнение расчетов в Си-программе 17](#_Toc198252461)

[d) Режимы работы: 18](#_Toc198252462)

[e) Обработка результатов в Bash 19](#_Toc198252463)

[f) Связи программы с другими программами 21](#_Toc198252464)

[8) Вызов и загрузка 22](#_Toc198252465)

[8.1) Способ вызова программы с соответствующего носителя данных 22](#_Toc198252466)

[8.2) Входные точки в программу 22](#_Toc198252467)

[9) Входные данные 24](#_Toc198252468)

[9.1) Характер и организация входных данных 24](#_Toc198252469)

[10) Выходные данные 26](#_Toc198252470)

[10.1) Характер и организация выходных данных 26](#_Toc198252471)

[10.2) Формат и кодирование выходных данных 27](#_Toc198252472)

[10.3) Пример файла massiv\_Uvx.txt: 27](#_Toc198252473)

[10.4) Пример консольного вывода таблицы: 27](#_Toc198252474)

[11) Структура кода 28](#_Toc198252475)

[12) Заключение 30](#_Toc198252476)

[13) Список используемой литературы 31](#_Toc198252477)

[14) Сокращения 32](#_Toc198252478)

[15) Приложения 34](#_Toc198252479)

[15.1) Приложение 1 34](#_Toc198252480)

[15.2) Приложение 2 39](#_Toc198252481)

[a) Файлы с исходным кодом 39](#_Toc198252482)

[b) Заголовочные файлы 40](#_Toc198252483)

[15.3) Приложение 3 42](#_Toc198252484)

# Задание к курсовой работе

Работа посвящена решению задач машинного анализа электрических цепей.

В курсовой работе необходимо для заданной электрической цепи по известному входному сигналу определить выходной сигнал для N равностоящих моментов времени, а затем определить некоторые его характеристики с погрешностью не более 1%. Варианты параметров входного сигнала (код А) и передаточной характеристики (код Б) электрической цепи приведены в приложении. Номер варианта определяется преподавателем индивидуально для каждого студента.

|  |  |
| --- | --- |
| **Входной сигнал** | **Рабочий набор** |
|  |  |

|  |  |
| --- | --- |
| **Выходной сигнал** | **Рабочий набор** |
|  |  |

* В ходе работы необходимо:
  + Произвести расчет входного и выходного сигнала в контрольных точках, используя при этом математический пакет Wxmaxima;
  + Написать текст программы на языке Си;
  + Произвести запись полученных результатов в файлы данных;
  + Используя математический пакет Wxmaxima (электронные таблицы), построить графики зависимости напряжений входных и выходных сигналов от времени.
  + Объединить программу на Си и Wxmaxima (LibraOffice.Calc), вызов отчета с помощью скрипта на Bash.

# Аннотация

Программный продукт "Анализатор сигналов электрических цепей" представляет собой консольное приложение, разработанное для автоматизированного анализа характеристик линейных электрических цепей. Программа выполняет численное моделирование прохождения сигнала через электрическую цепь с заданными параметрами.

**Основные функциональные возможности:**

* Расчет выходного сигнала по известному входному воздействию
* Определение временных параметров сигнала (длительность импульса, время нарастания)
* Оценка точности вычислений с заданной погрешностью (до 1%)

**Технические особенности реализации:**

* Язык разработки: С (ядро вычислений) + Bash (интерфейс)
* Платформа: ОС Linux (Ubuntu)
* Графическая подсистема: wxMaxima
* Способ взаимодействия: командная строка (Command line interface)

**Ключевые алгоритмы:**

* Дискретизация временной оси
* Кусочно-линейная аппроксимация
* Итерационный метод уточнения параметров
* Автоматическое построение графиков

Программа разработана в соответствии с требованиями ЕСПД (ГОСТ 19.ххх) и предназначена для использования в учебном процессе и инженерных расчетах. Особенностью решения является сочетание высокой точности вычислений (использование 32-битной арифметики с плавающей точкой IEEE 754) с простотой использования через командный интерфейс.

Объем исходного кода: ~500 строк (без учета зависимостей)  
Требования к аппаратному обеспечению: процессор x86-64, 512 МБ ОЗУ, 10 МБ дискового пространства.

# Общие сведения

## Обозначение и наименование программы

Для корректной работы программа требует установленную русифицированную версию операционной системы Ubuntu Linux.  
Также необходима установка стороннего ПО **wxMaxima**. Установить его можно командой в терминале:

**sudo apt-get install wxmaxima**

Компилятор gcc обычно является встроенным в ОС Linux, однако при его отсутствии его можно установить последовательностью команд:

**sudo apt update**

**sudo apt install build-essential**

## Языки разработки:

Программа написана на языке программирования Си — на нём реализо-вана основная функциональность.

Меню реализовано с использованием Bash-скриптов, которые также запускают скрипт для wxMaxima, оформленный в виде текстового файла с расширением .mac.

# Функциональное назначение

## Решаемые задачи

Программа предназначена для численного и графического анализа сигналов в электрических цепях. Она решает следующие задачи:

* **Моделирование сигналов:**
* Расчёт функции входного напряжения Uвх(t), заданной в аналитической форме;
* Вычисление выходного напряжения Uвых(t), используя кусочно-линейную передаточную характеристику.
* Графическая визуализация:
  + Построение графиков Uвх(t) и Uвых(t) с помощью **wxMaxima**;
  + Экспорт полученных данных в форматы, совместимые с другими пакетами (например, GNU Plot или LibreOffice Calc).

## Назначение:

Программа **АСВЭЦ** предназначена для работы в среде **Ubuntu Linux**. Программа АСВЭЦ предназначена для работы в среде Ubuntu Linux.

Основное применение — образовательное: визуализация работы электрических цепей и сравнение различных численных методов анализа сигналов.

Также программа пригодна для инженерных целей — быстрой оценки параметров цепей с нелинейными элементами.

Проверка аналитических решений гарантирует точность вычислений. Дополнительно, программа поддерживает автоматизацию обработки результатов для различных наборов параметров.

## Ограничения в функциональности:

* **Совместимость с ОС:** программа работает только в Ubuntu Linux и не поддерживает Windows.
* **Ограничения по входным данным:**
  + - Временной диапазон жёстко зафиксирован: t ∈ [5, 60] (можно изменить вручную в коде);
    - Параметры цепи заданы для варианта №17/5;
    - Максимальное количество точек **Nmax = 10 000** (определено размером массива).
* **Численные ограничения:**
  + - Используется тип данных float, что ограничивает точность;
    - Итерационные методы могут не сойтись при слишком малом значении **eps**.
* **Системные требования:**
  + Только ОС **Ubuntu Linux**;
  + Наличие **wxMaxima** (для визуализации) и **gcc** (для компиляции) обязательно.
* **Ограничения пользовательского интерфейса:**
  + Отсутствие графического интерфейса — работа осуществляется через консоль;
  + Низкая устойчивость к некорректному вводу.
* **Примечание:** при необходимости использовать программу для других параметров, требуется вручную изменить настройки в исходном файле funct.c.

# Контрольный расчёт

|  |  |
| --- | --- |
| **Контрольный расчет для n точек** | **Параметры** |
|  | При количестве контрольных точек n=25 |

| **Расчёт параметра с заданной точностью** | **Параметры** |
| --- | --- |
|  | n = 10, eps = 1% |

|  |  |
| --- | --- |
| **График Uvx** | **Параметры** |
|  | n = 25 |

|  |  |
| --- | --- |
| **График Uvix** | **Параметры** |
|  | n = 25 |

# Используемые технические средства

Минимальные и рекомендуемые требования к техническим средствам, которые соответствуют программе «Анализ сигнала на выходе электрической цепи», указаны в таблице 4.1. А также bash выше 4 версии.

Таблица 5.1**:**

Требования к техническим средствам

|  |  |  |
| --- | --- | --- |
| **Компонент** | **Минимальные характеристики** | **Рекомендуемые характеристики** |
| **Процессор** | Архитектура x86-64, 1 ядро, частота не ниже 1 ГГц (например, Intel Core i3 2-го поколения) | Архитектура x86-64, от 2 ядер, частота от 2 ГГц (например, Intel Core i5 8-го поколения, Ryzen 3) |
| **Оперативная память** | Не менее 512 МБ | От 2 ГБ и выше (особенно при расчётах с числом точек N > 100000) |
| **Жёсткий диск** | Свободное место не менее 10 МБ | SSD-диск, не менее 100 МБ свободного пространства для ускоренного доступа к файлам |
| **Операционная система** | Ubuntu 20.04+, Debian 10+ или совместимые дистрибутивы Linux | Astra Linux или дистрибутивы Linux с GUI (например, GNOME, KDE) |
| **Дополнительное ПО** | - GCC версии не ниже 9.3.0- wxMaxima версии не ниже 20.06 | - GCC версии 12 и выше- wxMaxima версии 23.04 и выше |
| **Монитор** | Разрешение экрана не менее 1280×720 | Разрешение экрана Full HD (1920×1080) |
| **Графическая карта** | Интегрированная, не ниже Intel HD Graphics 4000 | - |

# Таблица идентификаторов

Таблица 6.1:

**Таблица идентификаторов**

|  |  |  |
| --- | --- | --- |
| **Переменная** | **Тип** | **Описание** |
| **n** | int | Количество точек разбиения |
| **eps** | float | Допустимая погрешность (для приближённого метода) |
| **a, b, c** | float | Коэффициенты линейных участков функции Uvx |
| **tn** | float | Начальное время |
| **t1, t2, t3, t4** | float | Моменты излома функции |
| **tk** | float | Конечное время |
| **Uvx1, Uvx2** | float | Пороговые значения для функции Uvix |
| **t[N]** | float | Массив временных точек |
| **Uvx[N]** | float | Массив значений входной функции |
| **Uvix[N]** | float | Массив значений выходной функции |

# Описание функций

## Таблица 6.1

**Таблица функций bash**

| **Функция** | **Описание** |
| --- | --- |
| **clear\_line()** | Очищает текущую строку в терминале и воспроизводит звуковой сигнал |
| **is\_number()** | Проверяет, является ли введённое значение числом, соответствующим регулярному выражению |
| **ts1()** | Выполняет контрольный расчёт для n точек, выводит таблицу значений t, Uvx, Uvix и сохраняет их в файл |
| **ts2()** | Выполняет расчёт параметра с заданной погрешностью eps, выводит таблицу с погрешностью и сохраняет её в файл |
| **out\_zast()** | Выводит заставку (ASCII-арт или логотип) из файла ./config/zast.txt |
| **Основной цикл while** | Управляет меню программы, обрабатывает выбор пользователя и вызывает соответствующие функции |

## Таблица 6.2

**Таблица функций Си**

| **Функция** | **Описание** |
| --- | --- |
| **run\_app()** | Главная функция приложения, инициализирует параметры и вызывает нужный метод (control\_calc, approx\_value, file\_out\_data) |
| **form\_time()** | Заполняет массив t временными точками с равномерным шагом от tn до tk |
| **form\_Uvx()** | Вычисляет массив Uvx по кусочно-линейному закону с изломами в точках t1, t2, t3, t4 |
| **form\_Uvix()** | Вычисляет массив Uvix как линейное преобразование Uvx (Uvix[i] = 2.5 \* Uvx[i] + 10) |
| **parametr()** | Вычисляет среднее значение массива U (используется в approx\_value) |
| **form\_tabl1()** | Выводит таблицу значений t, Uvx, Uvix в три строки (для control\_calc) |
| **control\_calc()** | Выполняет точный расчёт, заполняет массивы t, Uvx, Uvix и выводит таблицу |
| **approx\_value()** | Выполняет итеративный расчёт с удвоением n до достижения заданной погрешности eps |
| **file\_out\_data()** | Сохраняет массивы t, Uvx, Uvix в отдельные файлы (massiv\_t.txt, massiv\_Uvx.txt, massiv\_Uvix.txt) |

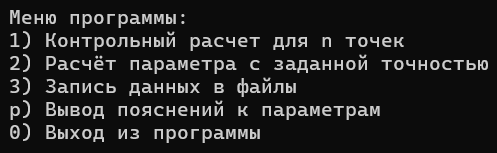
# Алгоритм работы программы

## Блок схема

## Описание логической структуры программы

### Инициализация и запуск

* Программа запускается через команду make run, которая вызывает главный скрипт menu.sh.
* menu.sh предоставляет пользователю интерактивное меню с вариантами:



### Обработка пользовательского ввода

#### При выборе варианта 1:

* Пользователь вводит n (количество точек, диапазон [2; 10000]).
* Bash-скрипт передает аргументы в Си-программу: ./prg 1 n.0

#### При выборе варианта 2:

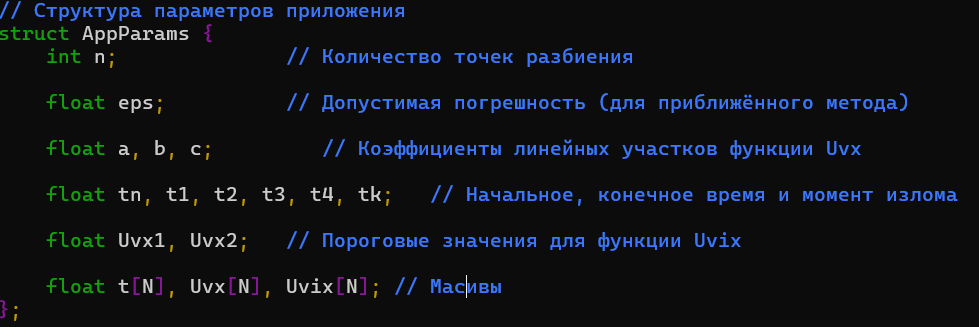
* Пользователь вводит n и eps (точность, диапазон [0.001; 99]).
* Bash-скрипт передает аргументы: ./myapp 2 n eps.

#### При выборе варианта 3:

* Вызывается си “./prg 3 n args” , где:
  + n – количество элементов в массиве
  + args - список элементов массивов: t/Uvx/Uvix
* Генерирует графики по массивам t/Uvx/Uvix

### Выполнение расчетов в Си-программе

* + Входные параметры передаются через структуру AppParams:

****

* + Основные функции:
* **form\_time():** Генерирует массив времени t с равномерным шагом.
  + **form\_Uvx():** Рассчитывает входное напряжение Uvx(t) как кусочно-линейную функцию.
  + **form\_Uvix():** Преобразует Uvx в выходное напряжение Uvix через пороговую аппроксимацию.
  + **parametr():** Вычисляет длительность превышения сигналом порогового значения.
  + **form\_tabl1():** Форматирует результаты в таблицу.

### Режимы работы:

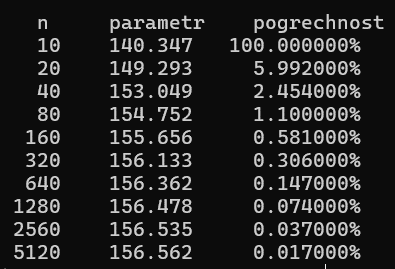
#### Вариант 1 (control\_calc):

* Вычисляет массивы **t, Uvx, Uvix** для заданного n.
* Возвращает таблицу значений в формате:



#### Вариант 2 (approx\_value):

* Итерационно увеличивает n (в 2 раза на каждом шаге) до достижения точности eps.
* Для каждого n вычисляет параметр duration (длительность превышения порога).
* Возвращает таблицу в формате:



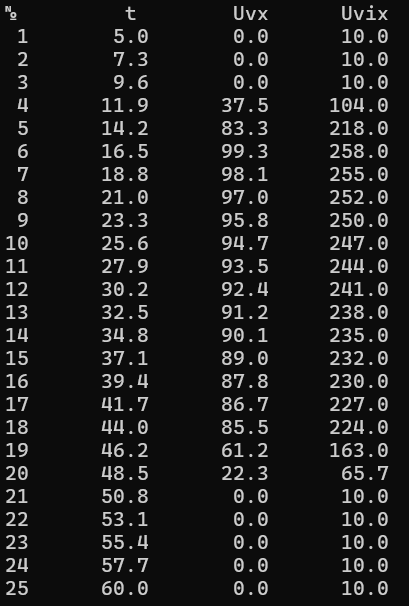
#### Вариант 3 (file\_out\_data):

* Формирует файлы из массивов t/Uvx/Uvix:
  + massiv\_t.txt
  + massiv\_Uvx.txt
  + massiv\_Uvix.txt

### Обработка результатов в Bash

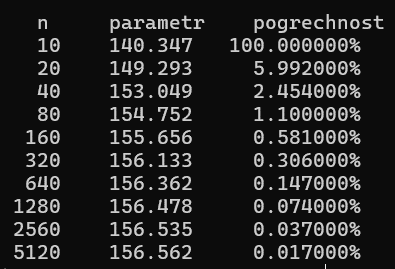
#### ****Для варианта 1****:

* Массивы выводятся виде таблицы:



#### Для варианта 2

* Итерационно увеличивает n (в 2 раза на каждом шаге) до достижения точности eps.
* Для каждого n вычисляет параметр duration (длительность превышения порога).
* Возвращает таблицу в формате:



#### Для варианта 3:

* Строятся графики через wxMaxima (скрипт make\_graphs.mac):
  + - graph\_Uvx.png — зависимость Uvx(t).
    - graph\_Uvix.png — зависимость Uvix(t).
* Пользователю предлагается просмотреть графики через eog.

| **График Uvx** | **График Uvix** |
| --- | --- |
|  |  |

#### Возврат в главное меню

После выполнения любого варианта программа возвращает пользователя в меню menu.sh для новых расчетов или выхода.

### Связи программы с другими программами

Программа АСВЭЦ в ходе своей работы запускает следующие стандартные для Linux библиотеки (способ установки указан в п. 1.2):

* wxmaxima
  + - * Используется для построения графиков.
      * Должна быть установлена в системе.
* GNU Compiler (gcc)
  + - * Обязателен для компиляции кода.

Все внешние вызовы выполняются через bash-скрипт menu.sh.

Программа работает в консоли, все вычисления выполняются последовательно, wxMaxima и gcc не являются частью программы. Программа также ориентирована на академические расчёты и локальное использование на Linux-системах.

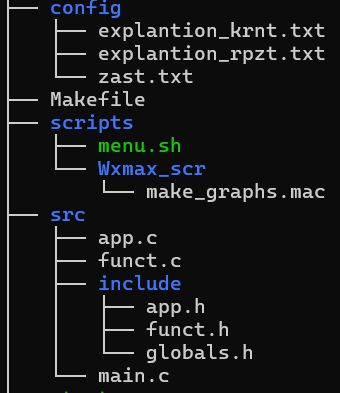
# Вызов и загрузка

## Способ вызова программы с соответствующего носителя данных

* Программа запускается вручную через терминал Linux следующим образом: make run
* Также возможен прямой вызов программы без меню, в случае его неработоспособности:
* Pg – это выбор контрольного расчёта/расчёт параметра
* n – кол-во элементов в массиве
* eps – предел точности погрешности
* args - список элементов массивов: t/Uvx/Uvix для 3 варианта
* ./myapp pg n eps args

## Входные точки в программу

Точкой входа в программу является главная функция – main() в файле main.c, которая вызывает функцию run\_app ().

Для работы программы необходимы установленные раннее пакеты (wxMaxima, gcc, eog), а также все обязательные файлы ().

Программа не требует прав суперпользователя (root), а все файлы данных для графиков сохраняются в текущую директорию.

# Входные данные

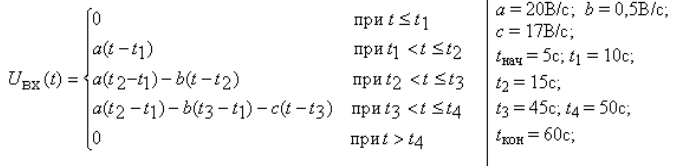
## Характер и организация входных данных

Программа АСВЭЦ (Автоматизированная Система Визуализации Электрических Цепей) использует два типа входных данных:

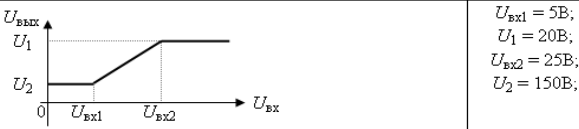
#### Фиксированные параметры цепи

Эти значения жёстко заданы в программе и не требуют ручного ввода:

* + - * **Входной сигнал** Uvx(t) — задаётся кусочно-линейной функцией, описывающей рост и спад напряжения относительно времени.



* + - * **Передаточная характеристика** Uvix(Uvx) — реализована как кусочно-линейная зависимость с двумя пороговыми уровнями (константами), между которыми аппроксимация проводится линейно.



#### ****Пользовательские параметры (вводятся через консоль)****

* **Количество точек** **N** — задаёт разрешение графика (число временных отсчётов).
* **Точность расчёта** **eps** — используется для приближённого метода (вариант 2), определяя относительную погрешность при расчёте параметра.
* **Подготовка входных данных не требуется** — все вспомогательные параметры и данные генерируются внутри программы автоматически.

#### Диапазоны допустимых значений:

* N ∈ [2, 10 000] — ограничение задано директивой #define N 15000.
* eps ∈ [0.001, 99.99] — значение вводится в процентах и преобразуется в доли (eps/100) внутри программы.

#### Пример ввода пользователем:

* Введите количество точек: 999
* Введите требуемую точность: 0.01

#### ****Кодировка:****

Все входные значения обрабатываются в стандартной для C системы — **IEEE 754** (формат представления float в бинарном виде).

# Выходные данные

## Характер и организация выходных данных

Программа формирует два основных вида выходных данных:

#### Текстовые файлы с результатами расчётов

* massiv\_t.txt — значения временной сетки t.
* massiv\_Uvx.txt — значения сигнала Uvx.
* massiv\_Uvix.txt — значения выходного сигнала Uvix.

Файлы автоматически сохраняются в папке ./data/, по одному значению в строке. Также отдельно сохраняется таблица в table\_p1.txt или table\_p2.txt (в зависимости от варианта).

#### Консольный вывод

* Интерфейс реализован с цветной разметкой (поддержка tput, ANSI escape codes).
* Отображаются подсказки, результаты расчётов и таблицы.
* Возможность вызова вспомогательного окна xterm для вывода пояснений.
* Поддержка запуска графиков в графическом интерфейсе (eog).

#### Графики

* Генерация графиков происходит через **Maxima** (скрипт make\_graphs.mac).
* Графики сохраняются в ./data/graphs/ и могут быть открыты через eog.

| **График Uvx** | **График Uvix** |
| --- | --- |
|  |  |

## Формат и кодирование выходных данных

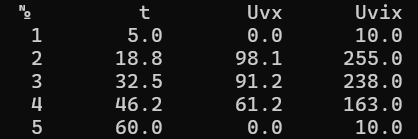
* **Формат текстовых файлов** — обычный текст UTF-8 (одна строка = одно значение).
* **Формат чисел** — вывод значений осуществляется с точностью до 6 знаков после запятой.
* **Кодировка** — UTF-8, стандартная для Linux-терминала.

## **Пример файла** massiv\_Uvx.txt**:**

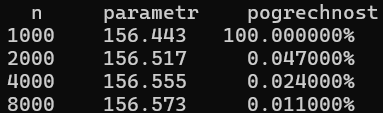
* 0.000000
* 1.800000
* 3.600000

## Пример консольного вывода таблицы:

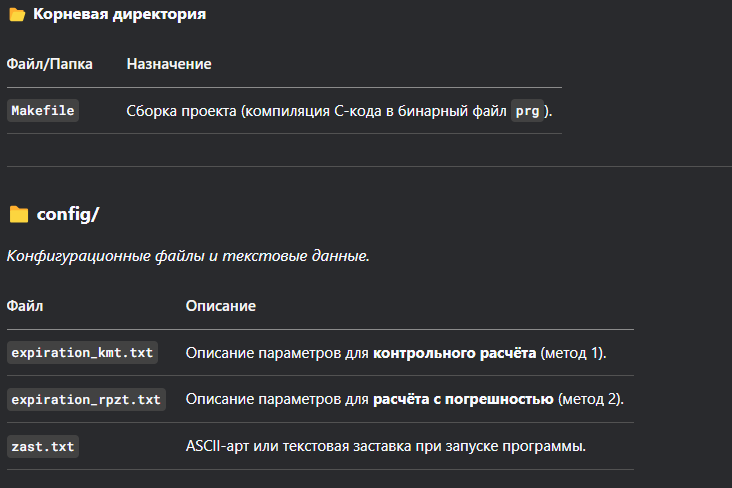
#### Контрольный расчет для n точек

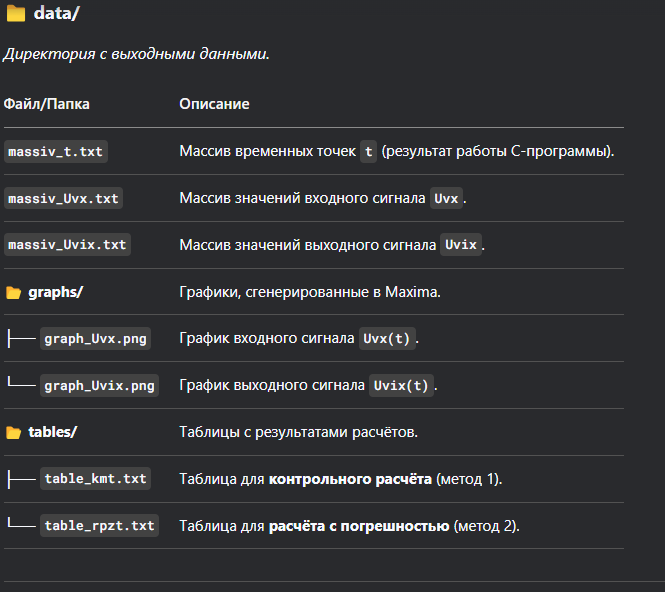


#### Расчёт параметра с заданной точностью

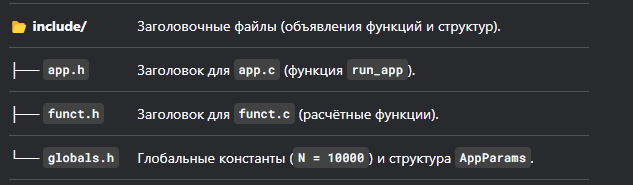


# Структура кода









Код Cи находится в приложение 1.

Код Bash находится в приложение 2.

Код Wxmaxima находится в приложение 3.

# Заключение

Практические данные, полученные в ходе работы программы, совпали с теоретическими расчетами, что подтверждает корректность реализованного алгоритма и достоверность результатов.

В результате выполнения проекта я успешно реализовал программу для моделирования и анализа электрических сигналов, сочетая математические методы, алгоритмизацию и автоматизацию процессов.

**Основные достижения:**

* **Разработал программный комплекс на C и Bash, включающий:**
  + Моделирование сигналов с заданной точностью
  + Автоматическую обработку и визуализацию данных
  + Гибкий интерфейс для управления расчетами
* **Освоил ключевые технологии:**
  + Численные методы (аппроксимация, погрешность 1%)
  + Работу с математическими библиотеками
  + Инструменты визуализации (Maxima, GNUPlot)
* **Систематизировал процесс разработки:**
  + Четкое разделение кода на модули (C/Bash)
  + Автоматизированную генерацию отчетов и графиков
  + Контроль достоверности результатов

**Итог:**

Проект позволил закрепить навыки в области computational science, от теоретического моделирования до практической реализации. Полученный опыт можно применять в задачах анализа данных, автоматизации измерений и инженерных расчетах.

# Список используемой литературы

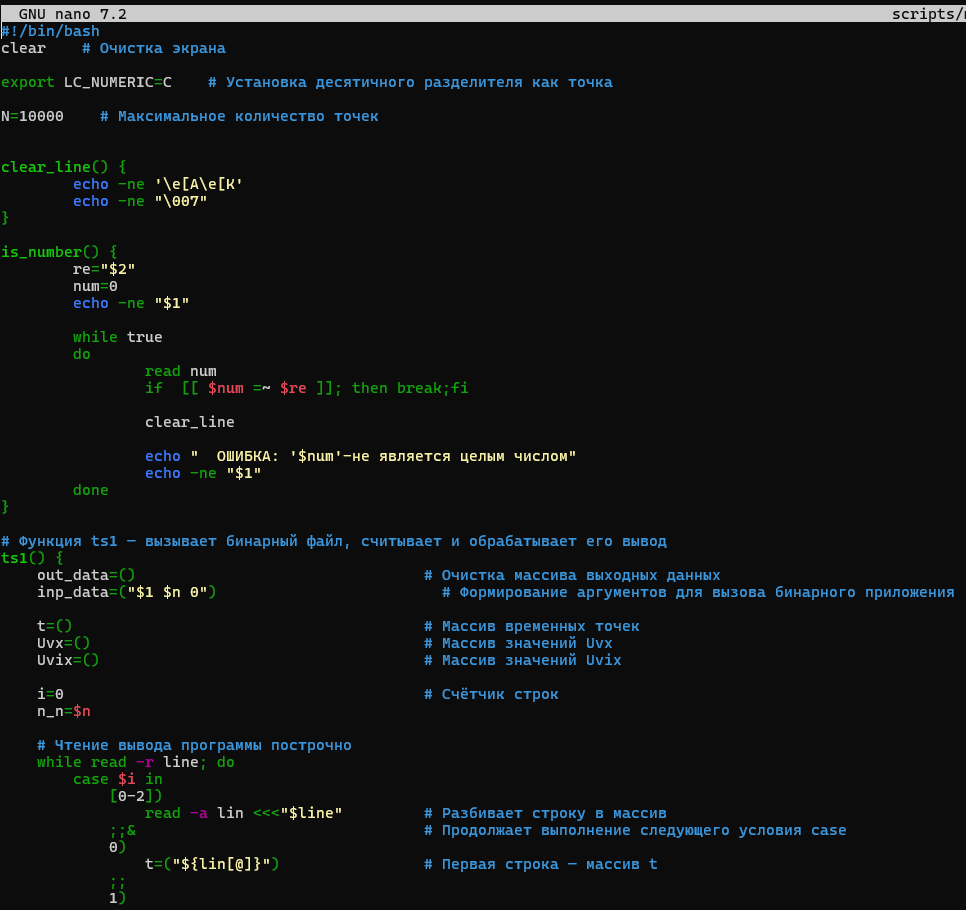
1. ГОСТ 19.402-78. Единая система программной документации. Пояснительная записка.
2. . ГОСТ 19.701-90. ЕСПД. Схемы алгоритмов, программ, данных и систем.
3. Брауде Э.Я. Основы программирования на языке C. — М.: Финансы и статистика.
4. Документация GNU Bash. URL: <https://www.gnu.org/software/bash/>
5. Документация wxMaxima. URL: <https://wxmaxima-developers.github.io/wxmaxima/>
6. Документация Си. URL: <https://c-language-documentation.vercel.app/>

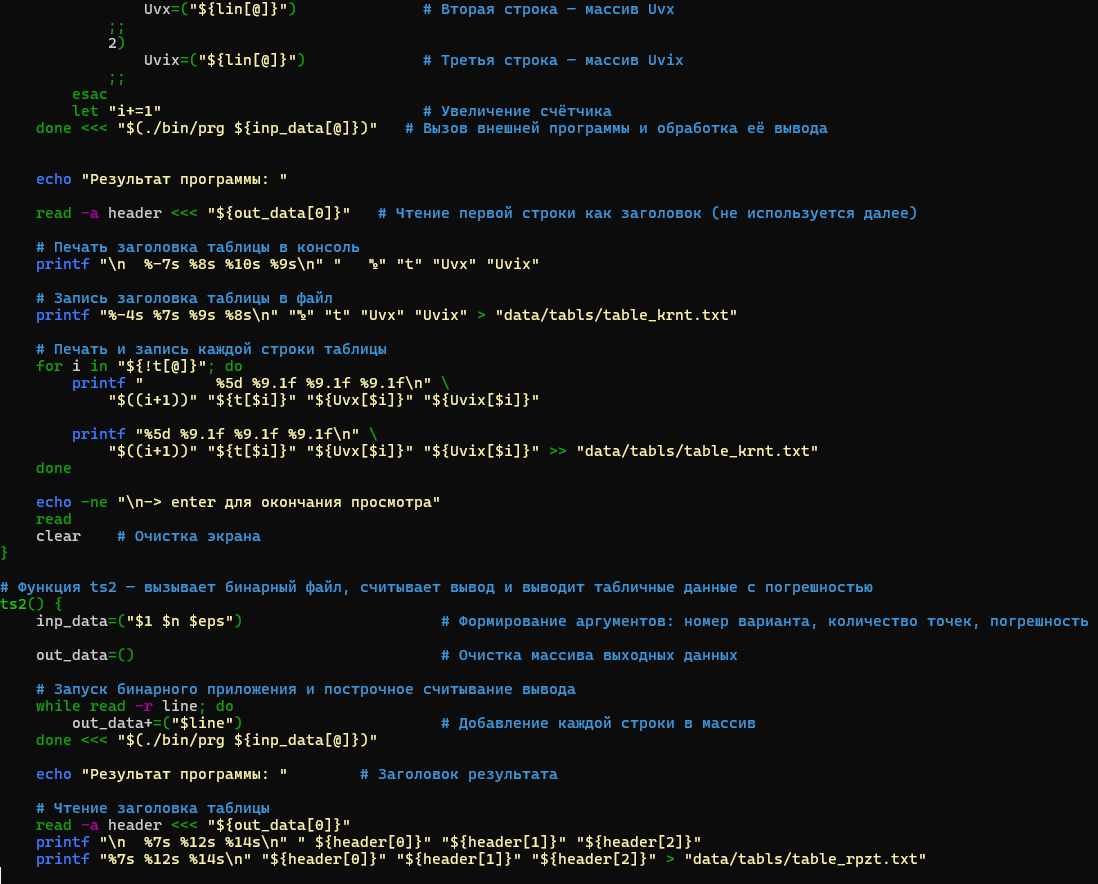
# Сокращения

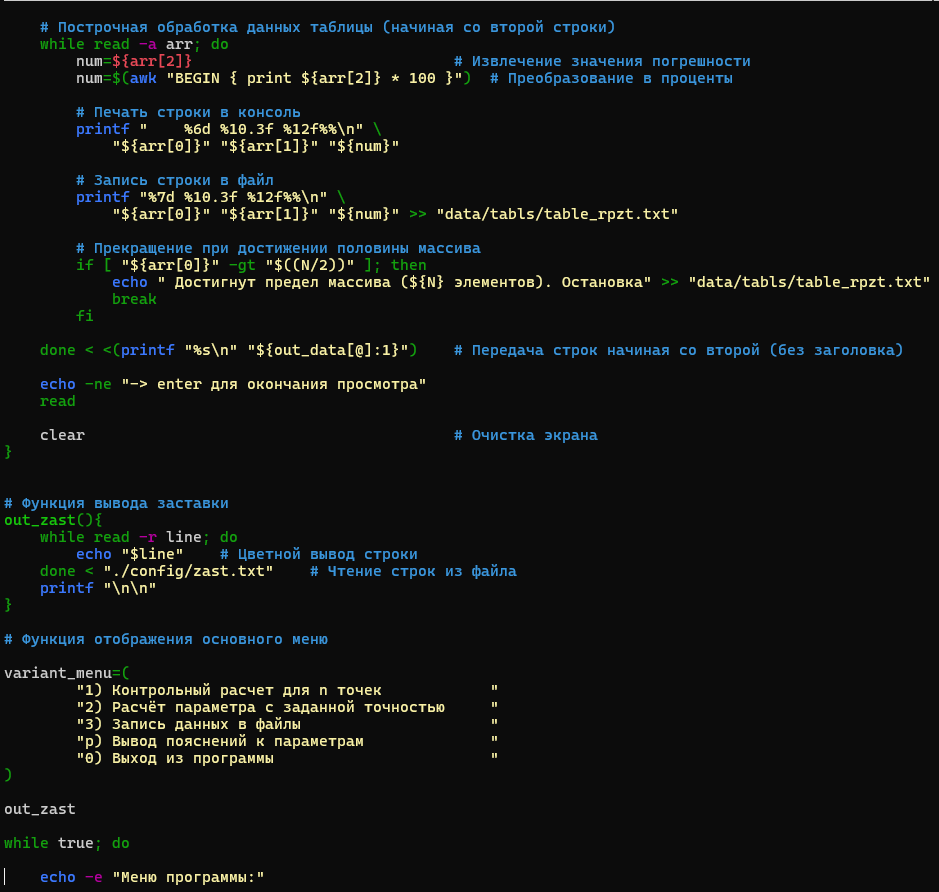
1. **ГОСТ- Г**осударственный **О**бщесоюзный **СТ**андарт
2. **URL** (Uniform Resource Locator)  
   Унифицированный указатель ресурса — адрес веб-страницы или файла в интернете (например, https://example.com).
3. **ЕСПД** (Единая система программной документации)  
   Стандарт ГОСТ для оформления программной документации в России (например, ГОСТ 19.ххх).
4. **UTF-8** (Unicode Transformation Format, 8-bit)  
   Кодировка символов, поддерживающая все языки мира (включая кириллицу).
5. **ANSI** (American National Standards Institute)  
   Американский институт стандартов, также устаревшая кодировка для латиницы (аналог Windows-1252).
6. **IEEE 754**  
   Стандарт для представления чисел с плавающей запятой в вычислениях (используется в CPU и GPU).
7. **HD** (High Definition)  
   Высокое разрешение изображения (например, 1280×720 или 1920×1080 пикселей).
8. **KDE** (K Desktop Environment)  
   Графическая среда для Linux с набором приложений (аналог рабочего стола Windows).
9. **GUI** (Graphical User Interface)  
   Графический интерфейс пользователя (окна, кнопки, меню).
10. **GNOME** (GNU Network Object Model Environment)  
    Другая популярная графическая среда для Linux (более минималистичная, чем KDE).
11. **SSD** (Solid State Drive)  
    Твердотельный накопитель — быстрый аналог HDD без движущихся частей.
12. **МБ** (Мегабайт)  
    1 МБ = 1 048 576 байт (или 10⁶ байт в маркетинге).
13. **ГБ** (Гигабайт)  
    1 ГБ = 1024 МБ (объём памяти или хранилища).
14. **ГГц** (Гигагерц)  
    Единица частоты процессора (1 ГГц = 1 млрд тактов в секунду).
15. **АСВЭЦ**  
    Аббревиатура из вашей программы: «Анализ сигнала на выходе электрической цепи».
16. **ОС** (Операционная система)  
    Программное обеспечение для управления компьютером (Windows, Linux, macOS).
17. **ПО** (Программное обеспечение)  
    Любые программы и приложения (от ОС до текстовых редакторов).
18. **ЕСПД - Единая Система Программной Документации**

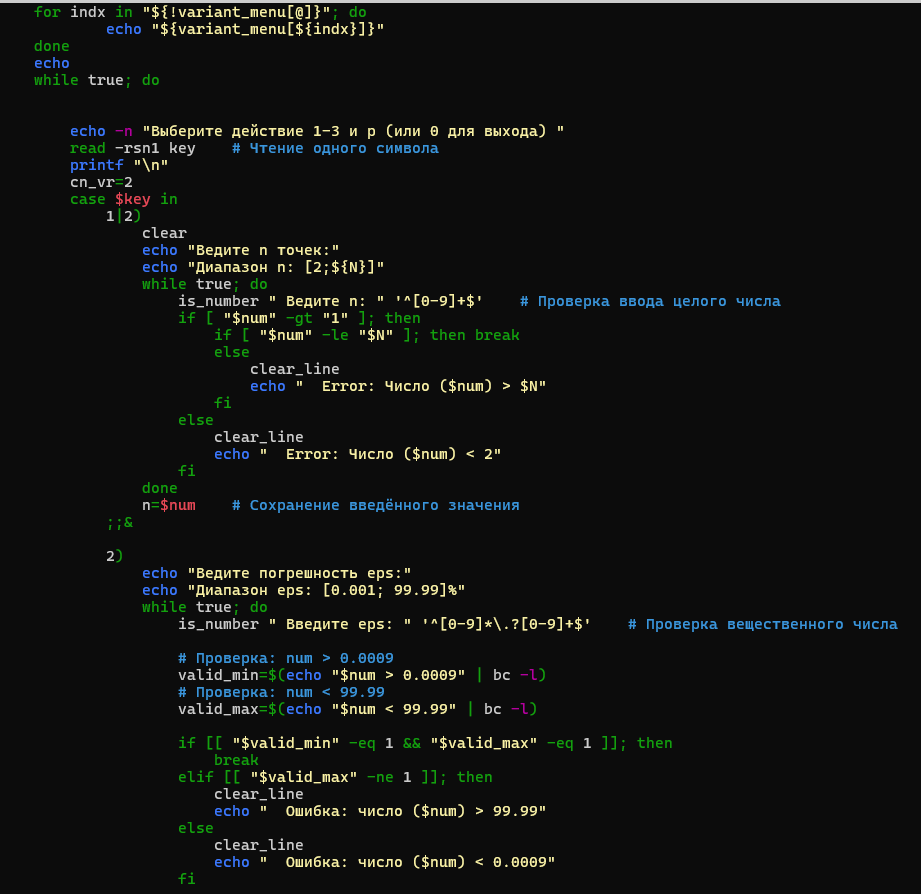
# Приложения

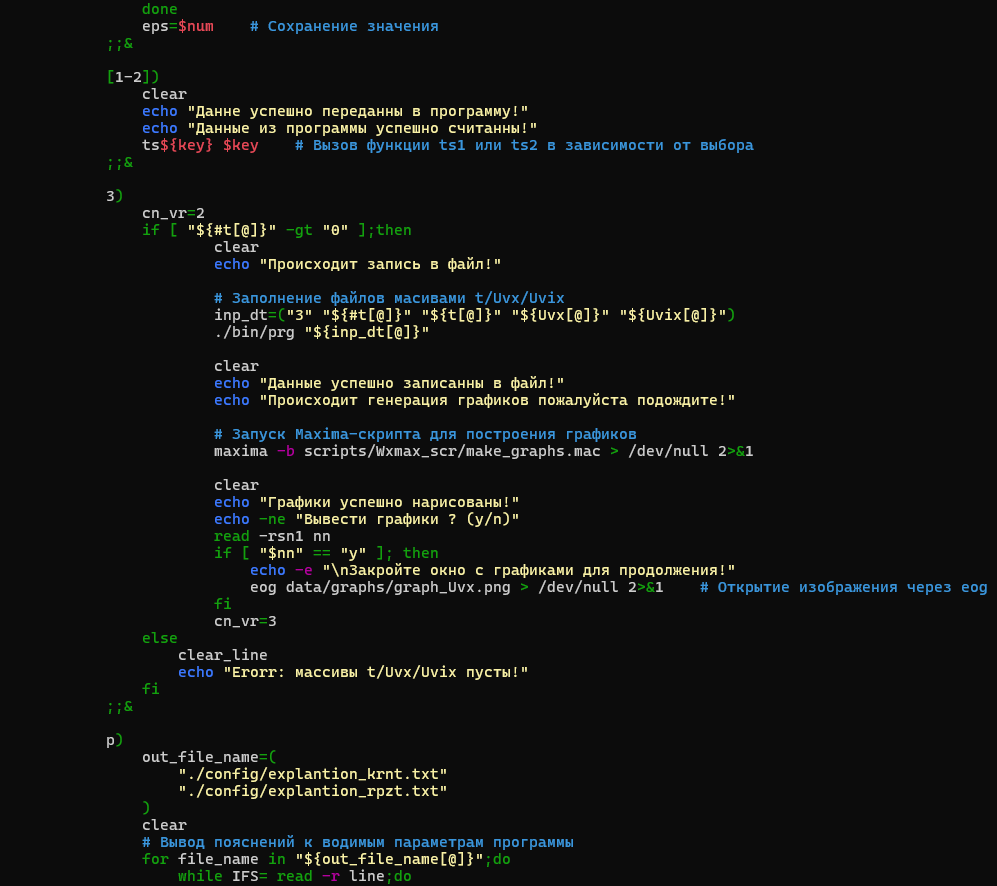
## Приложение 1

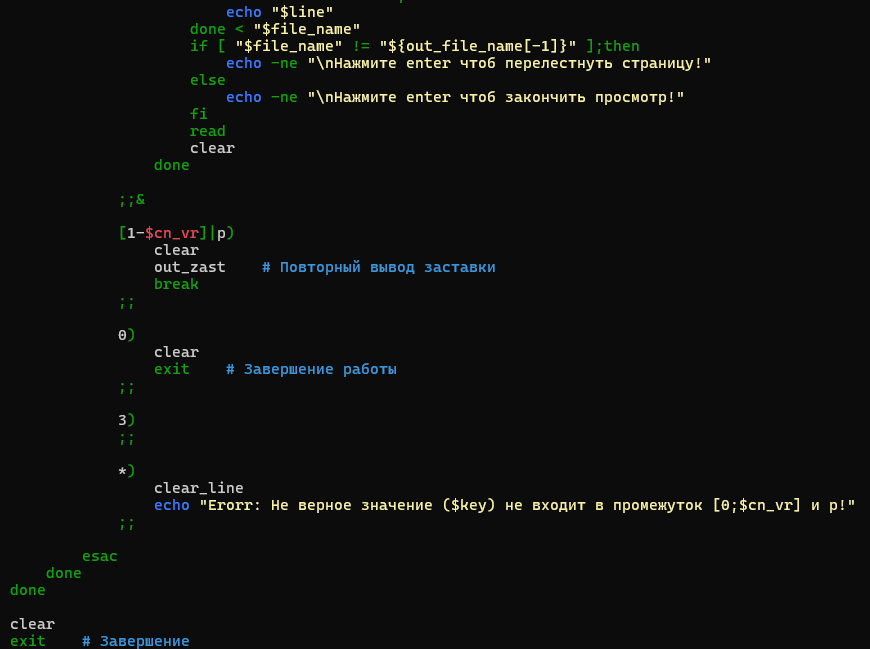






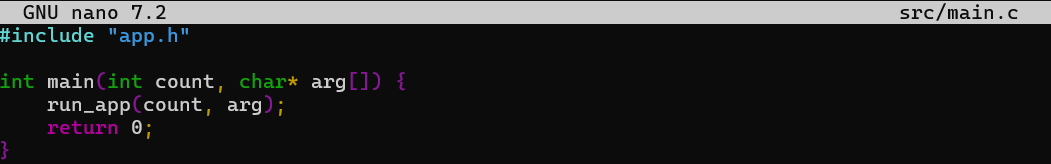


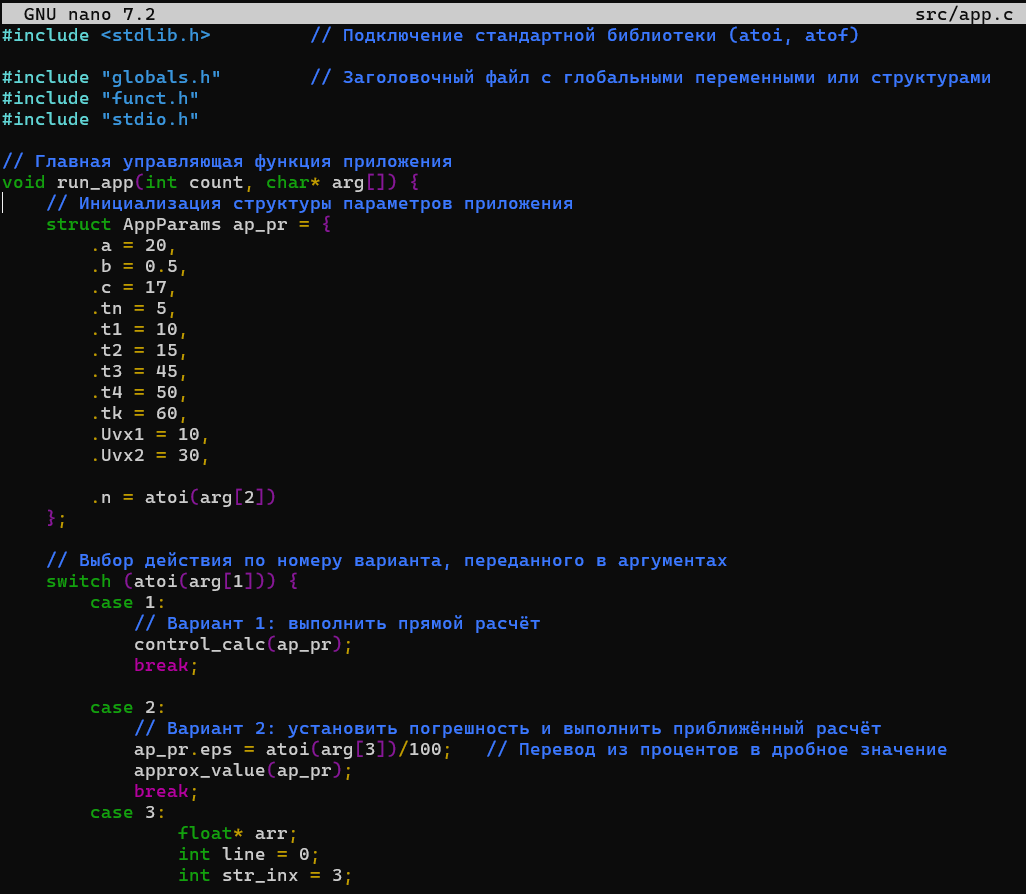


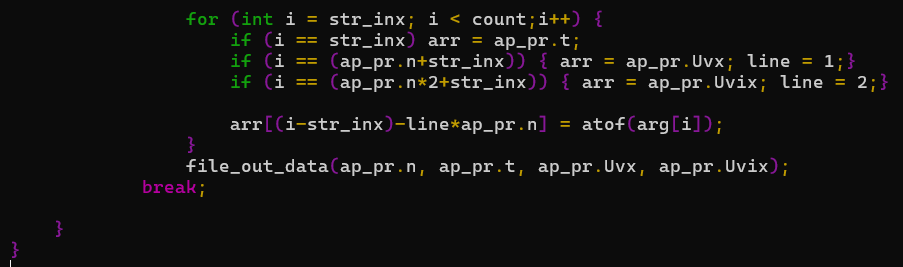


## Приложение 2

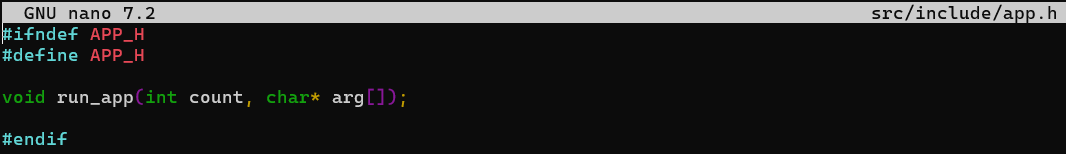
### Файлы с исходным кодом

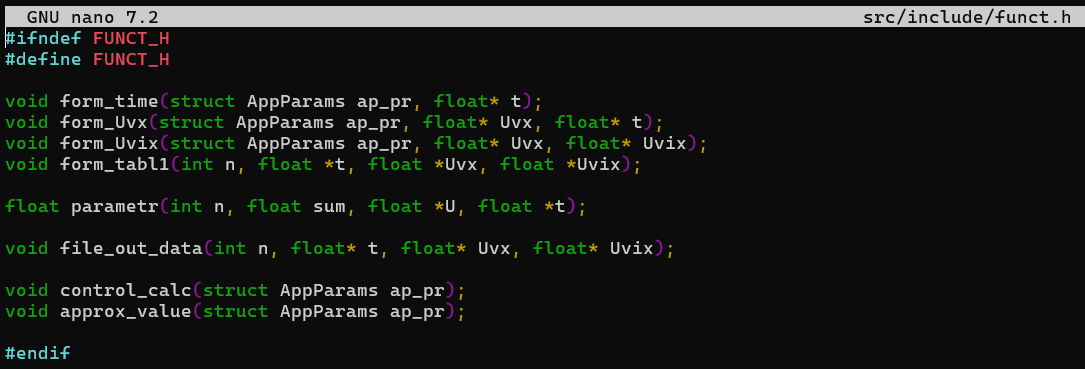


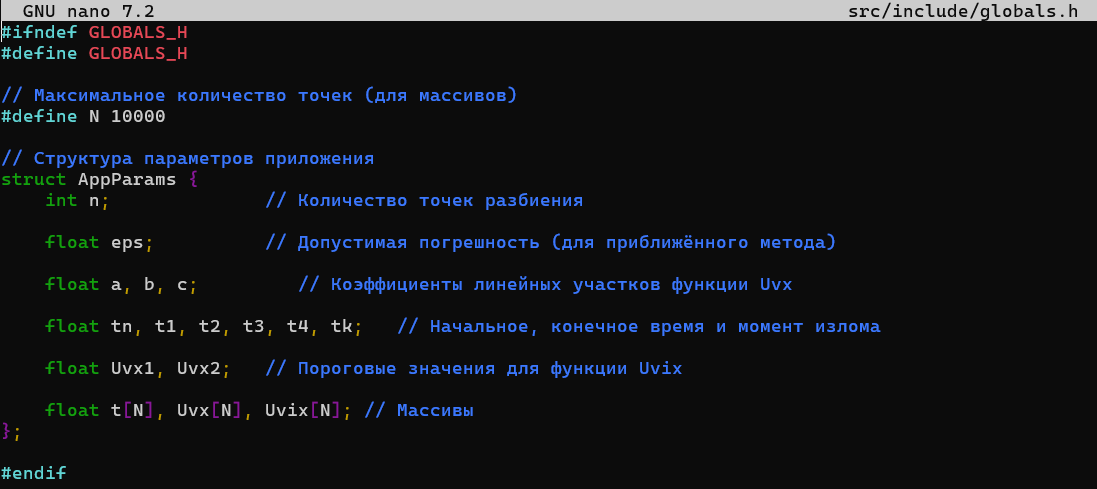




### Заголовочные файлы







## Приложение 3

