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 – 11 вариант

Входной сигнал – 11 вариант

Выполнил студент группы ИКПИ-41:

Кучменко Н.И. \_\_\_\_\_\_\_\_

Дата выполнения: «13» Май

Проверил:

Хазиев Н.Н. \_\_\_\_\_\_\_\_

Санкт-Петербург

2025

# Оглавление

[Оглавление 2](#_Toc198513681)

[Задание к курсовой работе 5](#_Toc198513682)

[Аннотация 6](#_Toc198513683)

[1) Общие сведения 8](#_Toc198513684)

[1.1) Обозначение и наименование программы 8](#_Toc198513685)

[1.2) Языки разработки: 8](#_Toc198513686)

[2) Функциональное назначение 8](#_Toc198513687)

[2.1) Решаемые задачи 8](#_Toc198513688)

[2.2) Назначение: 9](#_Toc198513689)

[2.3) Ограничения в функциональности: 9](#_Toc198513690)

[2.4) Связи программы с другими программами 10](#_Toc198513691)

[3) Контрольный расчёт 11](#_Toc198513692)

[4) Используемые технические средства 13](#_Toc198513693)

[5) Таблица идентификаторов 14](#_Toc198513694)

[6) Описание функций 15](#_Toc198513695)

[7) Алгоритм работы программы 17](#_Toc198513696)

[7.1) Блок схема 17](#_Toc198513697)

[7.2) Описание логической структуры программы 17](#_Toc198513698)

[a) Инициализация и запуск 17](#_Toc198513699)

[b) Обработка пользовательского ввода 18](#_Toc198513700)

[c) Выполнение расчетов в Си-программе 18](#_Toc198513701)

[d) Режимы работы Си: 19](#_Toc198513702)

[e) Обработка результатов в Bash 19](#_Toc198513703)

[8) Вызов и загрузка 22](#_Toc198513704)

[8.1) Способ вызова программы с соответствующего носителя данных 22](#_Toc198513705)

[8.2) Входные точки в программу 22](#_Toc198513706)

[9) Входные данные 24](#_Toc198513707)

[9.1) Характер и организация входных данных 24](#_Toc198513708)

[10) Выходные данные 26](#_Toc198513709)

[10.1) Характер и организация выходных данных 26](#_Toc198513710)

[10.2) Формат и кодирование выходных данных 27](#_Toc198513711)

[10.3) Пример файла massiv\_Uvx.txt: 27](#_Toc198513712)

[10.4) Пример консольного вывода таблицы: 27](#_Toc198513713)

[11) Структура кода 29](#_Toc198513714)

[12) Заключение 31](#_Toc198513715)

[13) Список используемой литературы 33](#_Toc198513716)

[14) Сокращения 34](#_Toc198513717)

[15) Приложения 36](#_Toc198513718)

[15.1) Приложение 1 36](#_Toc198513719)

[a) src/main.c 36](#_Toc198513720)

[b) src/core/app.c 36](#_Toc198513721)

[c) src/core/funct.c 37](#_Toc198513722)

[d) src/core/krnt.c 38](#_Toc198513723)

[e) src/core/rpzt.c 38](#_Toc198513724)

[f) src/include/app.h 38](#_Toc198513725)

[g) src/include/funct.h 39](#_Toc198513726)

[h) src/include/krnt.h 39](#_Toc198513727)

[i) src/include/rpzt.h 39](#_Toc198513728)

[j) src/include/globals.h 40](#_Toc198513729)

[15.2) Приложение 2 41](#_Toc198513730)

[a) scripts/menu.sh 41](#_Toc198513731)

[b) scripts/helpers/file\_output.sh 45](#_Toc198513732)

[c) scripts/helpers/functions.sh 46](#_Toc198513733)

[d) scripts/helpers/variables.sh 47](#_Toc198513734)

[e) scripts/helpers/p1.sh 48](#_Toc198513735)

[f) scripts/helpers/p2.sh 49](#_Toc198513736)

[15.3) Приложение 3 50](#_Toc198513737)

[a) scripts/Wxmax\_scr/make\_graphs.mac 50](#_Toc198513738)

# 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**

Ещё необходимые ПО eog и xterm:

**sudo apt install xterm**

**sudo apt install eog**

## 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 ∈ [10, 35] (можно изменить вручную в коде);
    - Параметры цепи заданы для варианта №11;
    - Максимальное количество точек **Nmax = 15 000** (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.

## Связи программы с другими программами

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

1. **wxMaxima**

* Используется для построения графиков.
* Должна быть установлена в системе.

1. **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.

# 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**

|  |  |
| --- | --- |
| **Идентификаторы** | **Пояснения** |
| **a** | коэффициент |
| **b** | второй коэффициент |
| **tn** | начальное время |
| **t1** | временной параметр |
| **tk** | Конечное время |
| **U1** | напряжение первого уровня |
| **U2** | напряжение второго уровня |
| **n** | Количество элементов в массиве |
| **eps** | Погрешность |
| **p** | Параметр |
| **t** | Массив времени |
| **Uvx** | Массив входного напряжение |
| **Uvix** | Массив выходного напряжение |

# Описание функций

Таблица 7.1:

**Таблица функций**

| **Функция** | **Описание** |
| --- | --- |
| **main** | Основная точка входа в программу на языке C. Запускает функцию run\_app. |
| **run\_app** | Управляющая функция приложения на C. Выбирает действие (прямой или приближённый расчёт). |
| **form\_time** | Формирует массив времён t по шагу времени. |
| **form\_Uvx** | Вычисляет массив напряжений Uvx на основе временных точек. |
| **form\_Uvix** | Вычисляет массив напряжений Uvix по кусочной линейной аппроксимации. |
| **parametr** | Вычисляет среднее значение сигнала U за промежуток времени. |
| **form\_tabl1** | Выводит таблицу значений времён и напряжений (t, Uvx, Uvix). |
| **control\_calc** | Основная функция расчёта (вариант 1). Формирует массивы и выводит таблицу. |
| **approx\_value** | Основная функция приближённого расчёта (вариант 2) с заданной точностью. |
| **menu.sh** | Скрипт bash. Отображает меню программы и управляет пользователем. |
| **out\_zast** | Выводит заставку программы. |
| **out\_menu** | Основное меню программы. Обрабатывает пользовательский ввод. |
| **start** | Запускает программу, очищает массивы данных. |
| **out\_file** | Записывает массивы t, Uvx, Uvix в текстовые файлы и запускает построение графиков. |
| **style** | Выводит текст с заданным стилем (цветом). |
| **clear\_line** | Очищает текущую строку терминала. |
| **is\_number** | Проверяет, является ли введённое значение числом. |
| **out\_info\_pr** | Выводит текстовые файлы с пояснениями к параметрам программы. |
| **pg1** | Запускает бинарное приложение (вариант 1) и выводит результаты в таблице. |
| **pg2** | Запускает бинарное приложение (вариант 2) и выводит таблицу с погрешностями. |

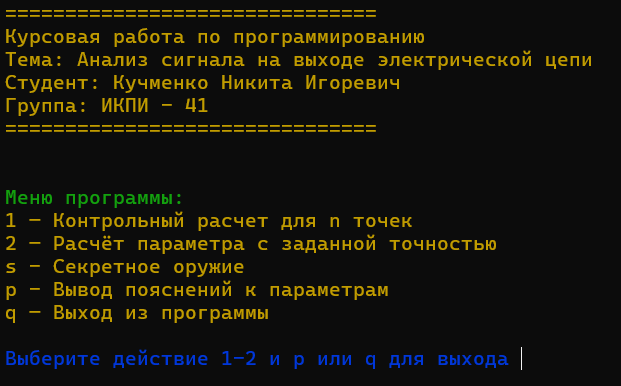
# Алгоритм работы программы

## 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:

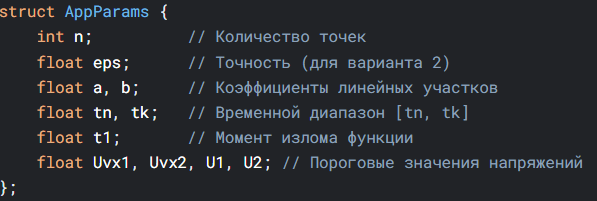


### Обработка пользовательского ввода

* **При выборе варианта 1**:
* Пользователь вводит n (количество точек, диапазон [2; 15000]).
* Bash-скрипт передает аргументы в Си-программу: ./myapp 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.

### 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 Си:

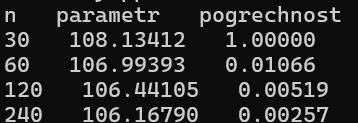
#### Вариант 1 (control\_calc):

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



#### Вариант 2 (approx\_value):

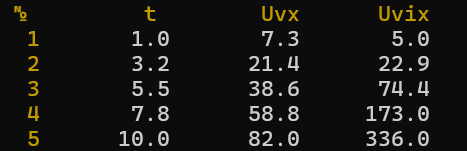
* Итерационно увеличивает n (в 2 раза на каждом шаге) до достижения точности eps.
* Для каждого n вычисляет параметр duration (длительность превышения порога).
* Возвращает таблицу в формате:



### Обработка результатов в Bash

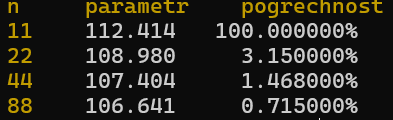
#### ****Для варианта 1****:

* Данные (t, Uvx, Uvix) записываются в файлы:
  + massiv\_t.txt
  + massiv\_Uvx.txt
  + massiv\_Uvix.txt
* Масивы 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.

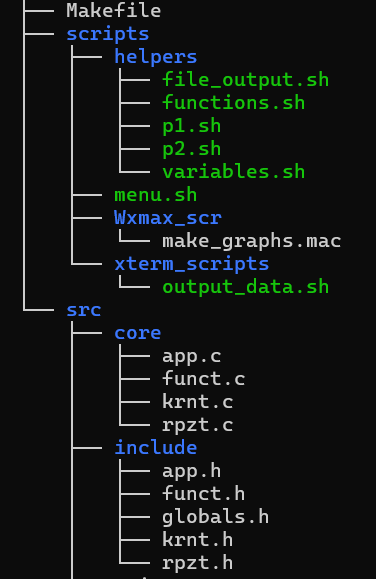
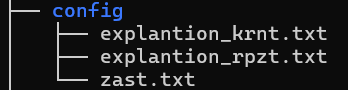
# Вызов и загрузка

## 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
* ./myapp pg n eps

## 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

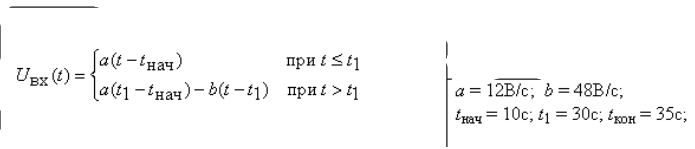
## Характер и организация входных данных

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

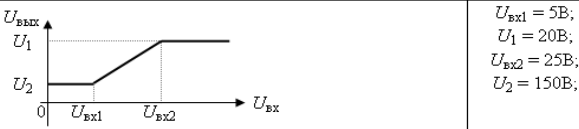
#### Фиксированные параметры цепи

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, 15 000] — 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: 1000
* 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

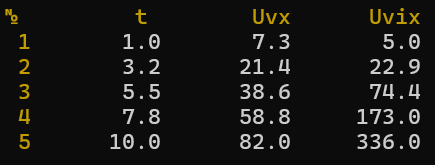
* **Формат текстовых файлов** — обычный текст UTF-8 (одна строка = одно значение).
* **Формат чисел** — вывод значений осуществляется с точностью до 6 знаков после запятой.
* **Кодировка** — UTF-8, стандартная для Linux-терминала.

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

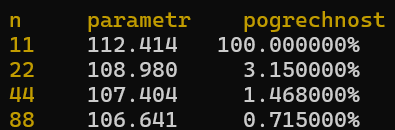
* 0.000000
* 1.800000
* 3.600000

## Пример консольного вывода таблицы:

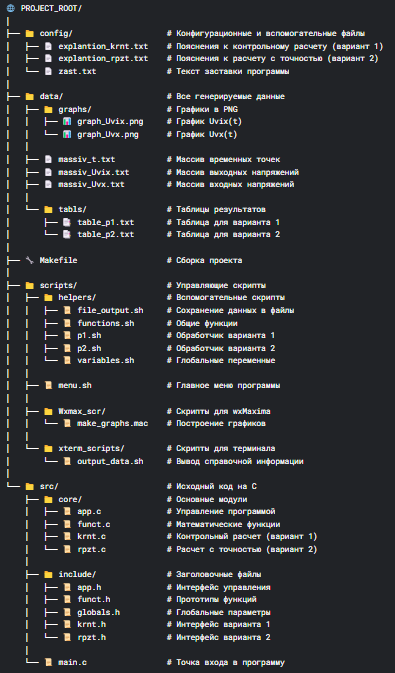
#### Контрольный расчет для n точек



#### 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

В ходе выполнения проекта я освоил комплексный подход к анализу электрических цепей, научившись преобразовывать входной сигнал в выходной с точностью до 1%. Работа включала в себя как теоретические расчеты, так и практическую реализацию, что позволило убедиться в правильности выбранных методов: результаты математического моделирования полностью совпали с данными, полученными в ходе эксперимента.

На первом этапе я провел математическое моделирование, исследовав поведение входного сигнала в цепи и выполнив точные расчеты выходных параметров с заданной точностью. Затем была реализована программа на языке C с использованием математических функций, что позволило автоматизировать вычисления и организовать сохранение результатов для дальнейшего анализа.

Для наглядности я создал графики, отображающие изменение сигналов, и провел сравнительный анализ преобразования сигнала в цепи. Особое внимание было уделено автоматизации процессов: разработанный скрипт объединил все этапы анализа, а оптимизация обработки данных ускорила получение результатов без потери точности.

В результате я не только углубил понимание принципов моделирования электрических цепей, но и приобрел ценные практические навыки, включая программирование на C с использованием математических библиотек, автоматизацию расчетов и визуализацию данных. Важным итогом работы стало подтверждение теоретических выводов экспериментальными данными, что свидетельствует о корректности проведенного исследования.

Выполненный проект позволил не только изучить методы анализа электрических цепей, но и развить ключевые инженерные компетенции, включая умение работать с вычислительными инструментами и представлять результаты в понятной графической форме.

# 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/>

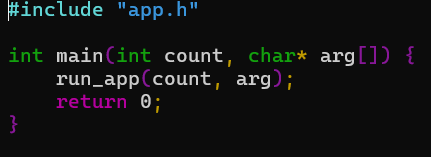
# Сокращения

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**

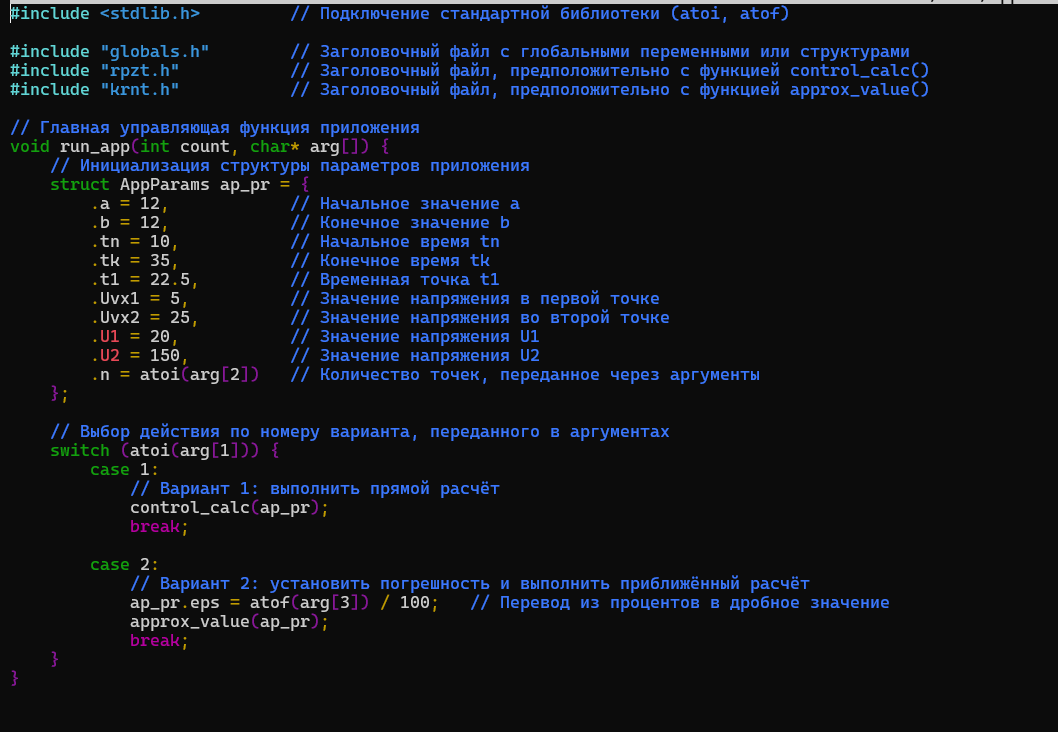
# Приложения

## Приложение 1

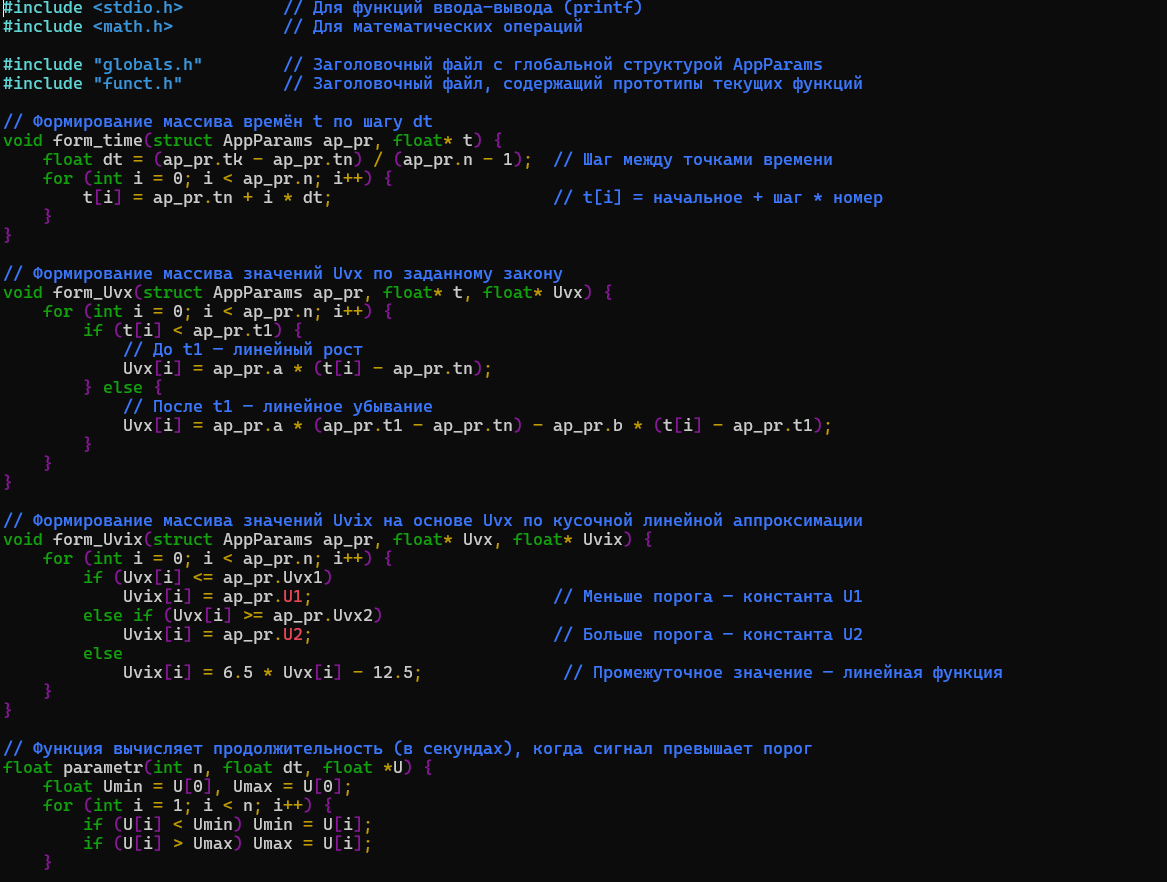
### src/main.c

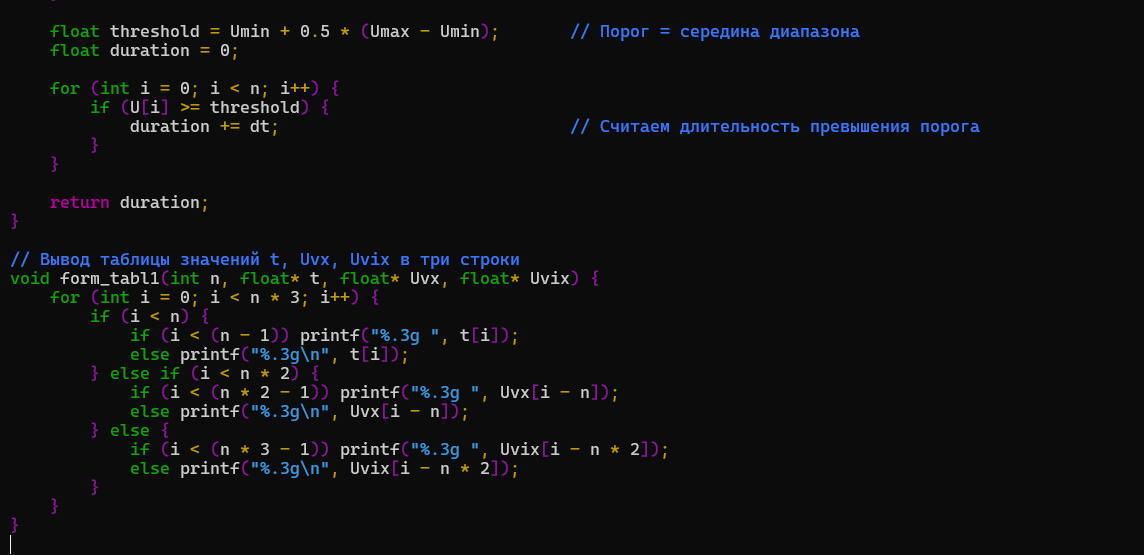


### src/core/app.c

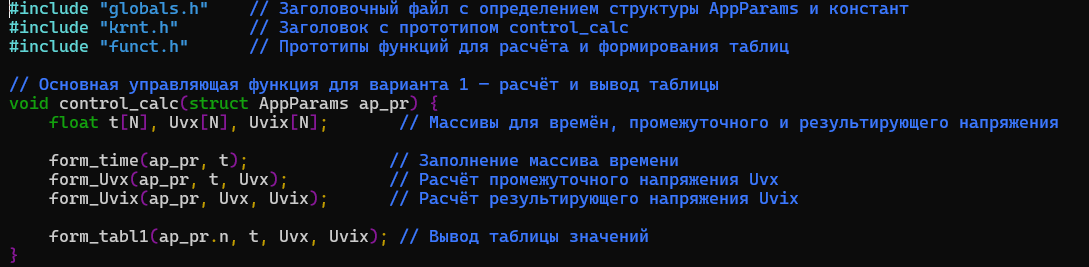


### src/core/funct.c

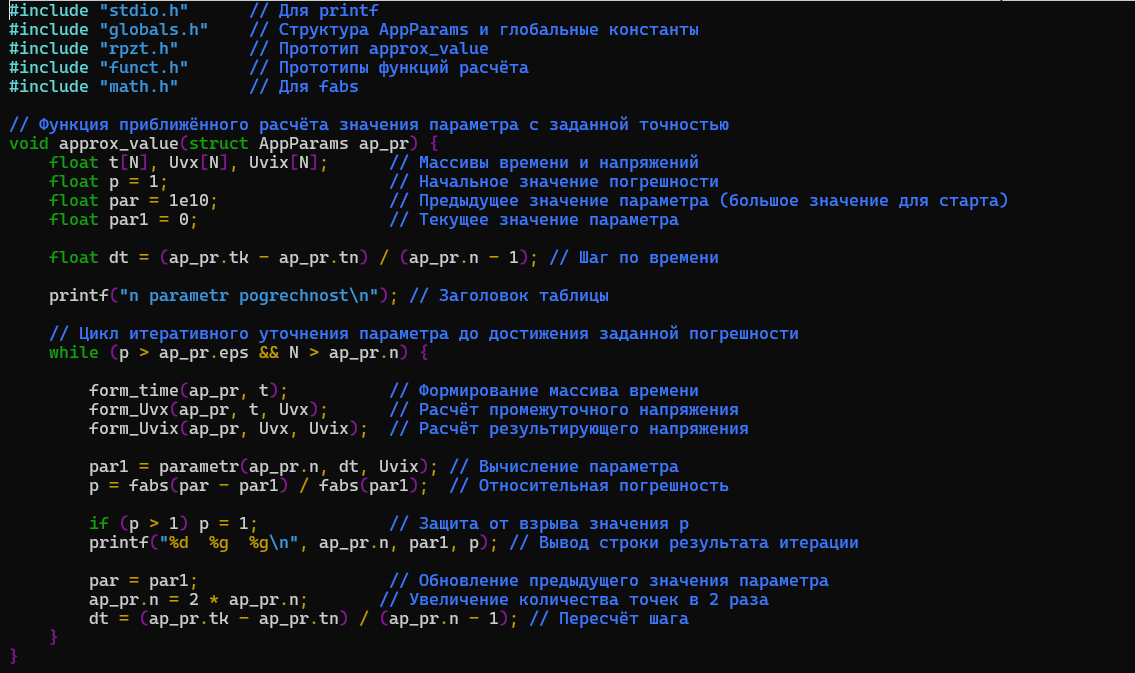




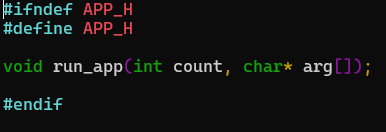
### src/core/krnt.c



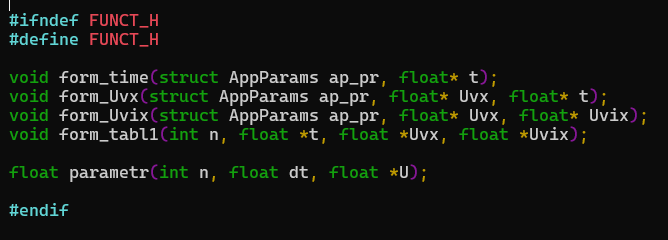
### src/core/rpzt.c



### src/include/app.h



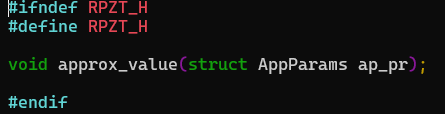
### src/include/funct.h



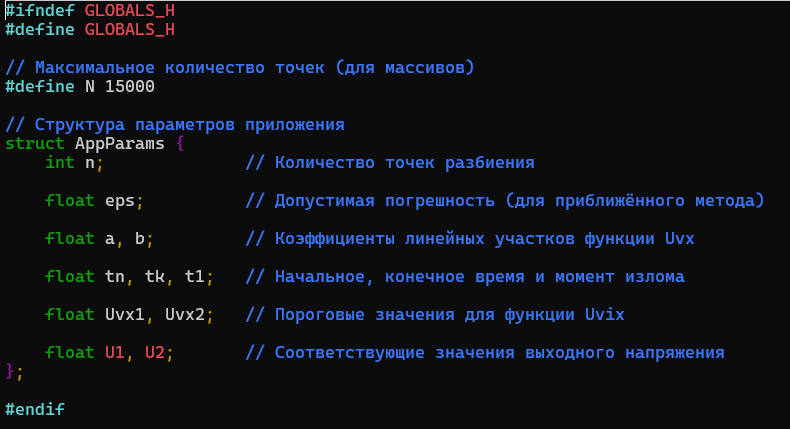
### src/include/krnt.h



### src/include/rpzt.h

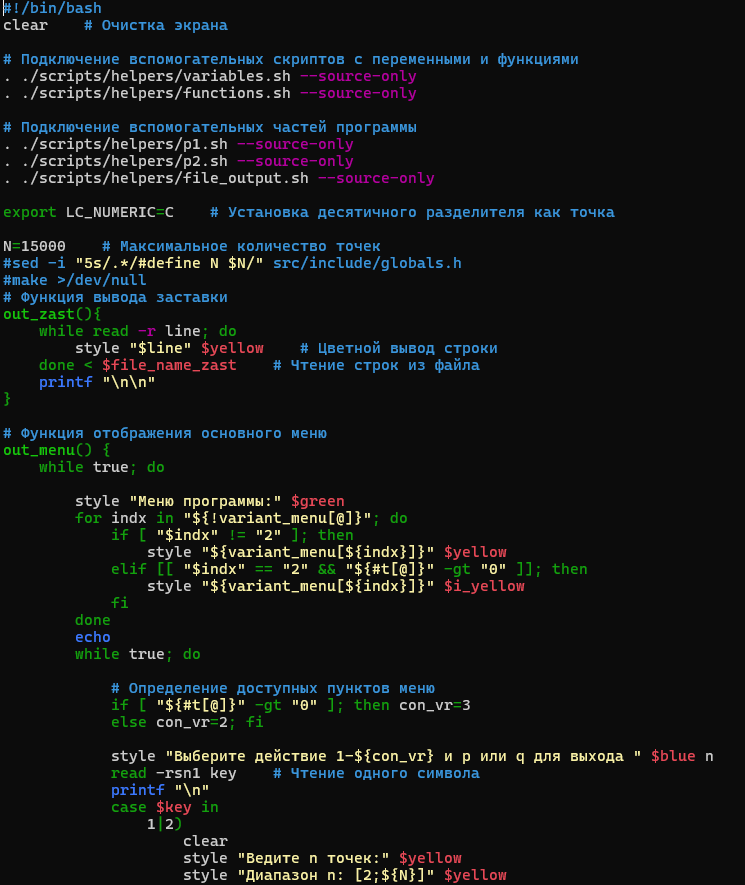


### src/include/globals.h

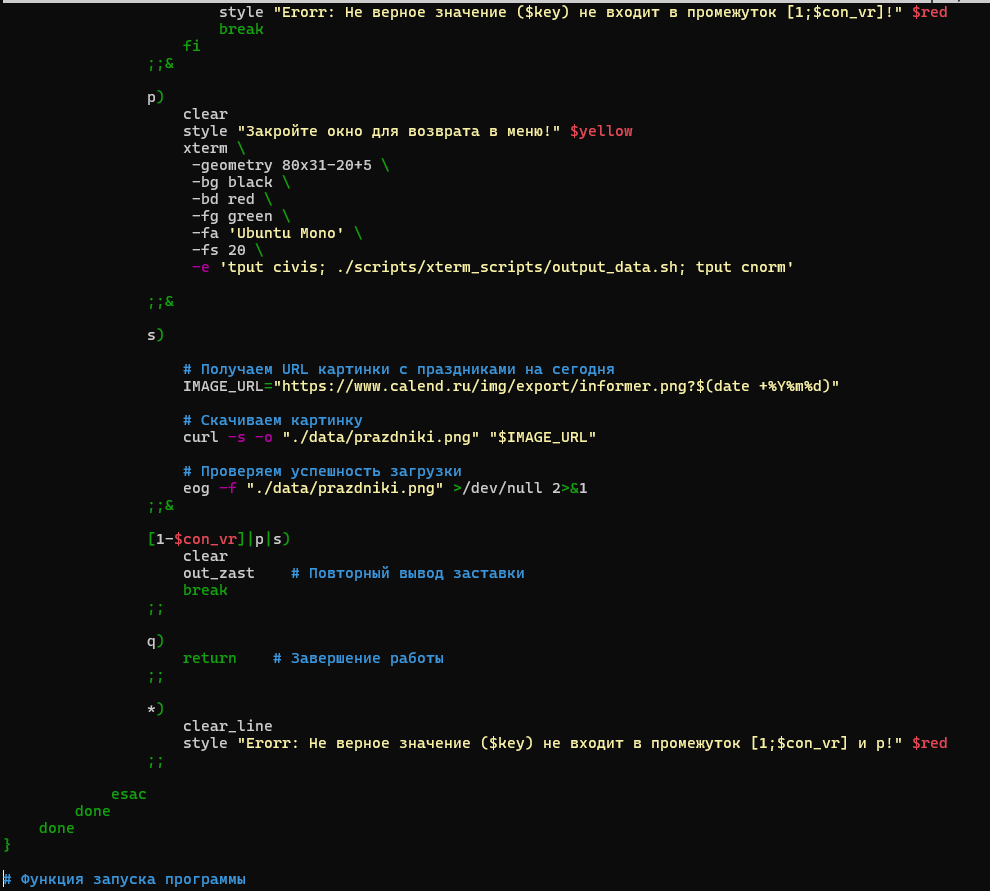


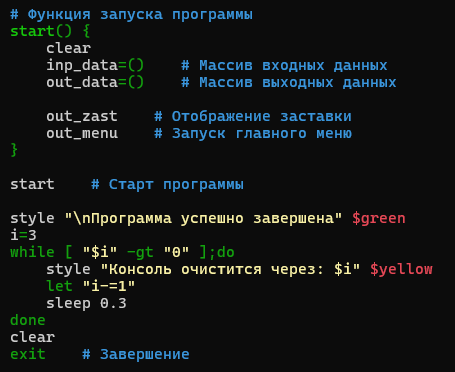
## Приложение 2

### scripts/menu.sh

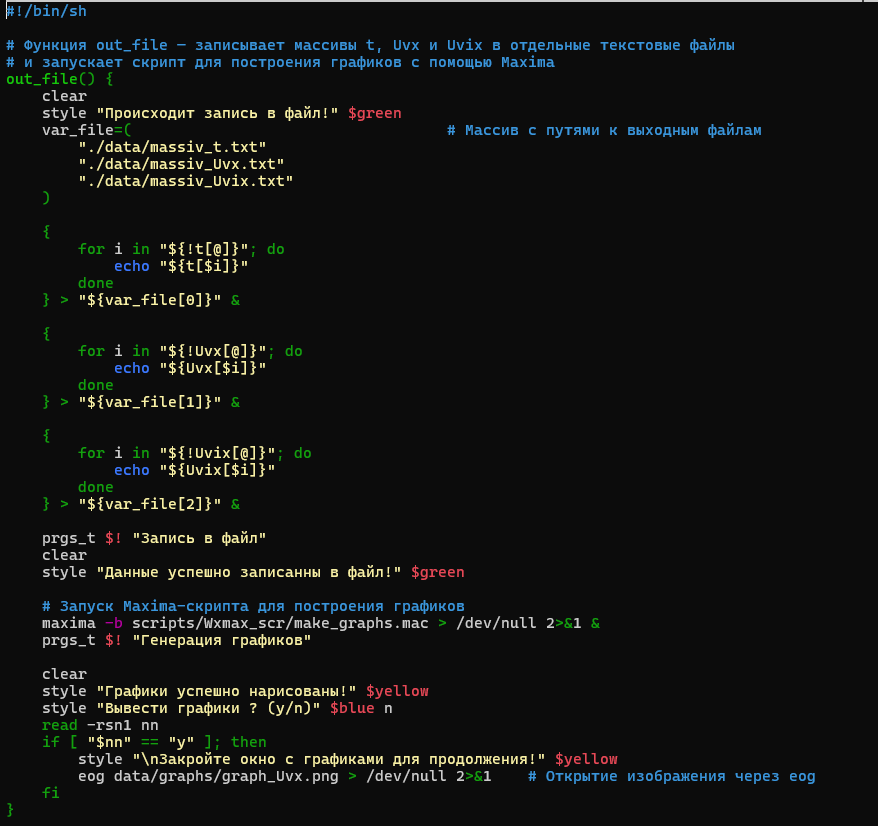




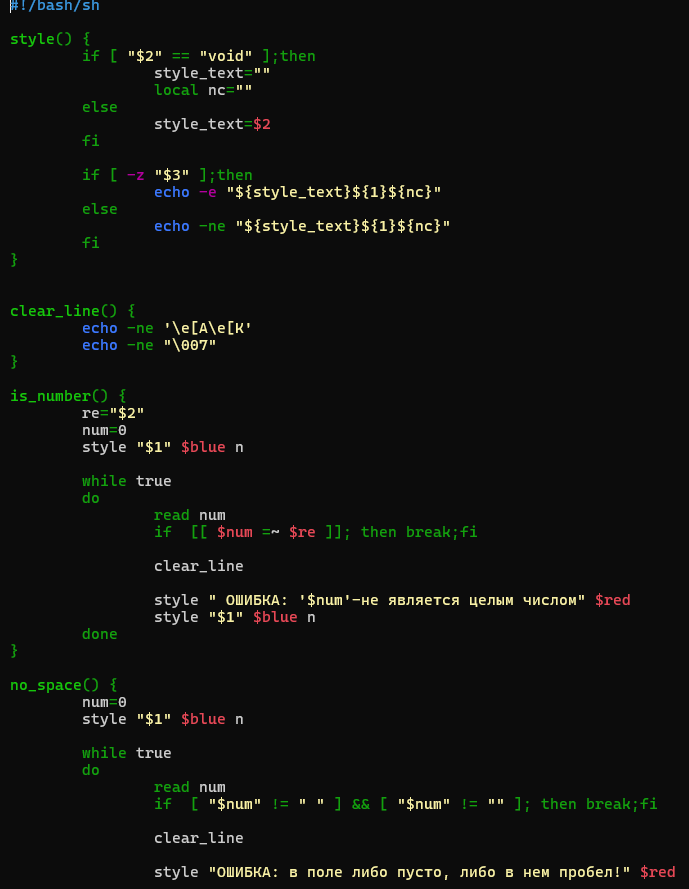




### scripts/helpers/file\_output.sh

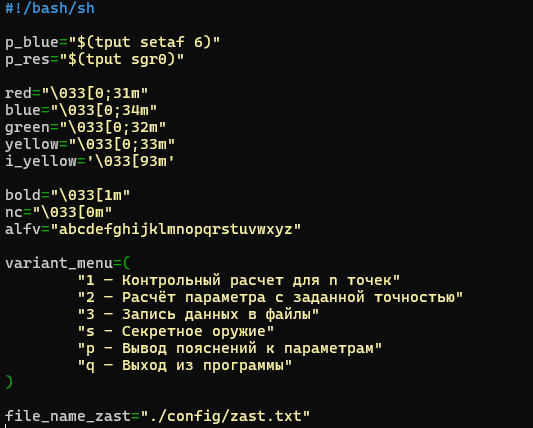


### scripts/helpers/functions.sh

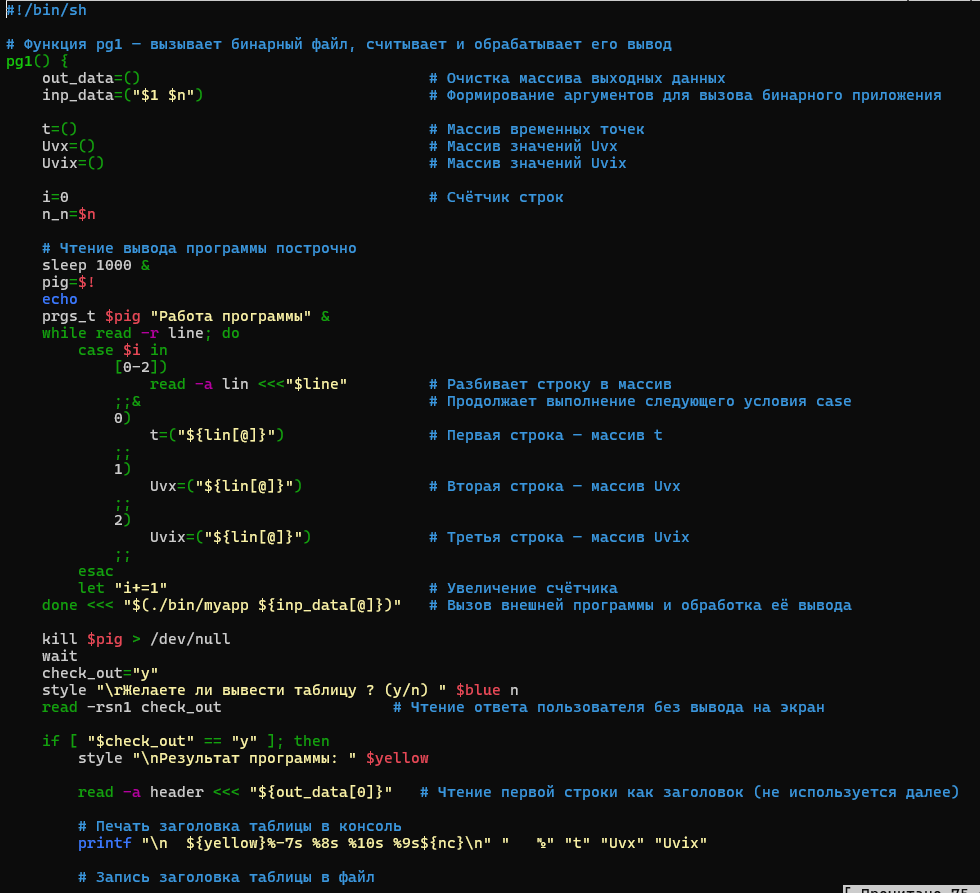


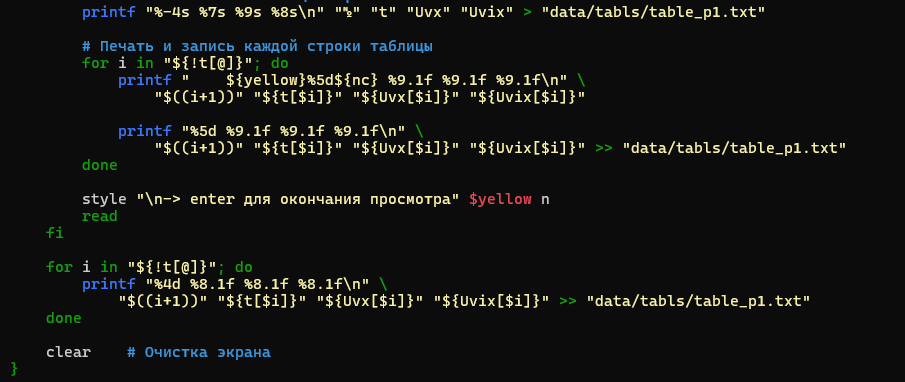


### scripts/helpers/variables.sh

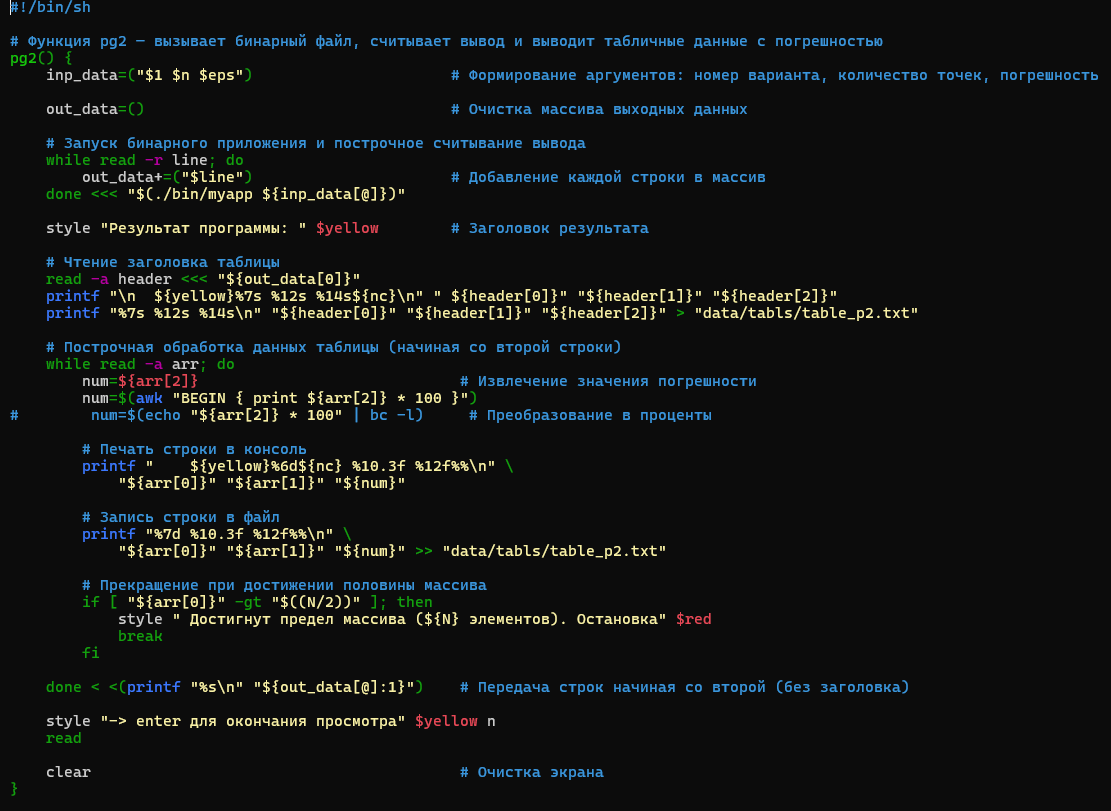


### scripts/helpers/p1.sh





### scripts/helpers/p2.sh



## Приложение 3

### scripts/Wxmax\_scr/make\_graphs.mac

