

# НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ «ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

## Отчёт по домашней работе

«Вычисление квадратного корня с заданной точностью с использованием итерационного метода»

Студент: Ошаров Александр

Семинарист: Кензин Игорь

Дата: 2025-10-10

---

### 1. Цель работы

Реализовать программу на языке ассемблера RISC-V, вычисляющую квадратный корень из положительного действительного числа с заданной точностью с использованием итерационного метода. Программа должна включать модульную структуру: отдельные файлы для основной логики, подпрограммы вычисления корня и макроопределений, а также обеспечивать корректную обработку входных данных и вывод результата.

---

### 2. Используемый алгоритм

Для вычисления квадратного корня применяется **метод Ньютона (метод касательных)**, также известный как **метод Герона**. Итерационная формула имеет вид:

$$x_{n+1} = \frac{1}{2} \left( x_n + \frac{a}{x_n} \right)$$

где:

- (  $a > 0$  ) — исходное число,
- (  $x_0 = a$  ) — начальное приближение,
- Вычисления продолжаются до тех пор, пока  $|x_{n+1} - x_n| < \varepsilon$ , где  $\varepsilon > 0$  — заданная точность.

Метод обладает квадратичной сходимостью и гарантирует корректный результат при  $a > 0$  и  $\varepsilon > 0$ .

---

### 3. Структура программы

Программа состоит из трёх файлов:

- `main.s` — главный модуль, содержащий точку входа и вызов макросов.
- `sqrt.s` — подпрограмма `sqrt_iter`, реализующая итерационный алгоритм.
- `macros.s` — файл с макроопределениями для ввода, проверки, вызова и вывода.

Такой подход обеспечивает чёткое разделение ответственности и упрощает сопровождение кода.

---

### 4. Описание компонентов

#### 4.1. Макроопределения ( `macros.s` )

| Макрос                     | Назначение   |
|----------------------------|--|
| <code>INPUT_DATA</code>    | Считывает с консоли число ( <code>a</code> ) и точность ( <code>ε</code> ) (два вызова <code>read_double</code> ). |
| <code>CHECK_INPUT</code>   | Проверяет, что оба значения строго положительны. При нарушении — вывод ошибки и завершение.                        |
| <code>CALL_SQRT</code>     | Вызывает подпрограмму <code>sqrt_iter</code> для вычисления корня.   |
| <code>OUTPUT_RESULT</code> | Выводит строку-заголовок и результат ( <code>print_double</code> ).  |

Все макросы инкапсулируют логику, не используя глобальные переменные, и работают через стандартные соглашения о вызовах RISC-V (аргументы в `fa0`, `fa1`; результат в `fa0` ).

#### 4.2. Подпрограмма вычисления корня ( `sqrt.s` )

Функция `sqrt_iter`:

- Принимает аргументы: `fa0 = a`, `fa1 = ε`.
- Использует регистры с плавающей точкой ( `ft0 – ft9` ) для промежуточных вычислений.

- Реализует цикл по формуле Ньютона до достижения заданной точности.
- Возвращает результат в  $\sqrt{a}$ .

---

## 5. Примеры тестирования

Ниже приведены примеры корректных и некорректных входных данных.

### 5.1. Корректные тесты

| Вход (a, ε)   | Ожидаемый результат ( $\sqrt{a}$ ) | Фактический результат |
|---------------|------------------------------------|-----------------------|
| 4.0, 0.001    | 2.0                                | 2.0000000929222947    |
| 2.0, 0.0001   | $\approx 1.4142$                   | 1.4142135623746899    |
| 100.0, 0.01   | 10.0                               | 10.000000000139897    |
| 0.25, 0.00001 | 0.5                                | 0.5000000000000006    |

Программа быстро сходится даже при малых значениях  $\epsilon$  благодаря квадратичной сходимости метода.

### 5.2. Некорректные тесты

| Вход (a, ε) | Реакция программы                |
|-------------|----------------------------------|
| -4.0, 0.001 | Вывод ошибки, завершение         |
| 4.0, -0.001 | Вывод ошибки, завершение         |
| 0.0, 0.001  | Вывод ошибки (0 не положительно) |

Во всех случаях программа корректно обнаруживает нарушение условий и завершает работу с сообщением об ошибке.

---

## 6. Скриншоты тестового запуска

**Примечание:** Скриншоты выполнения программы включены в приложение к отчёту (представлены отдельно).

Пример вывода в терминале:

```
2.0
0.0001
Корень квадратный: 1.4142135623746899
```

При вводе отрицательного числа:

```
-1.0
0.01
Ошибка: число и точность должны быть положительными!
```

---

## 7. Заключение

Работа успешно выполнена в соответствии с требованиями:

- Реализован итерационный метод Ньютона для вычисления квадратного корня.
- Обеспечена модульная структура программы с вынесением логики в отдельные файлы.
- Все макроопределения корректно инкапсулируют ввод, проверку, вызов и вывод.
- Программа устойчива к некорректному вводу и завершается с понятным сообщением об ошибке.
- Тестирование подтверждает корректность и точность вычислений.

Программа демонстрирует понимание принципов структурного программирования на ассемблере RISC-V, работы с числами с плавающей точкой и обработки исключительных ситуаций.

---

**Приложение: Исходный код содержится в текущем репозитории**