



# **Итоговая работа по курсу «Программирование на языке С (ПРОДВИНУТЫЙ уровень)»**

**Лымарь Дмитрий Олегович**

**(группа Д01-051)**

# Оглавление

Постановка задания	3
Описание программы	5
Исходный код программы	5
Описание файлов программы	5
Описание работы программы	6
Примеры написания командной строки для запуска:	6
Демонстрация работы программы	7
Сборка программы утилитой make	8

# Постановка задания

Составьте программу, которая находит площадь, образуемую данными уравнениями:

1.  $f_1 = 0.6x + 3$

2.  $f_2 = (x - 2)^3 - 1$

3.  $f_3 = 3 / x$

Требования к программе:

1. Основная программа должна поддерживать опции командной строки, при задании которых:

a. Печатаются абсциссы точек пересечения кривых

b. Печатается число итераций, потребовавшихся на приближённое решение уравнений при поиске точек пересечения

2. Программа должна поддерживать ключ командной строки `-help`, выводящий на печать все допустимые ключи командной строки.

3. Вычисление с точностью  $\epsilon_1$  корня  $x$  уравнения  $f(x) = g(x)$  на отрезке  $[a, b]$  должно быть реализовано в отдельной Си-функции `root(f, g, a, b, eps1)`. Если используется метод касательных или комбинированный метод, то у `root` должно быть ещё два параметра функционального типа, позволяющие вызывать производные функций  $f$  и  $g$

4. Вычисление с точностью  $\epsilon_2$  величины определённого интеграла от функции  $f(x)$  на отрезке  $[a, b]$  должно быть реализовано в отдельной Си функции `integral(f, a, b, eps2)`

5. Си-функции `root` и `integral` должны быть предварительно протестированы. Основная программа должна предоставлять возможности тестирования, активируемые опцией командной строки.

6. Сборка программы должна осуществляться при помощи утилиты `make`. Соответствующий файл должен явно или неявно описывать зависимости между всеми целями сборки. Должны быть определены цели `all` и `clean`, первая из которых полностью собирает программу, а вторая — удаляет все промежуточные файлы (в частности, объектные модули). Сдаваемый архив должен включать в себя `Makefile`

7. Программа должна быть снабжена поясняющими комментариями в объёме, достаточном для её понимания. Все глобальные и статические переменные должны быть документированы в комментариях.

## Решение через GeoGebra

На рис.1. показано вычисление площади фигуры полученное пересечением заданных фигур. Для этого нам нужно вычислить всего 2 интеграла от разности функций.

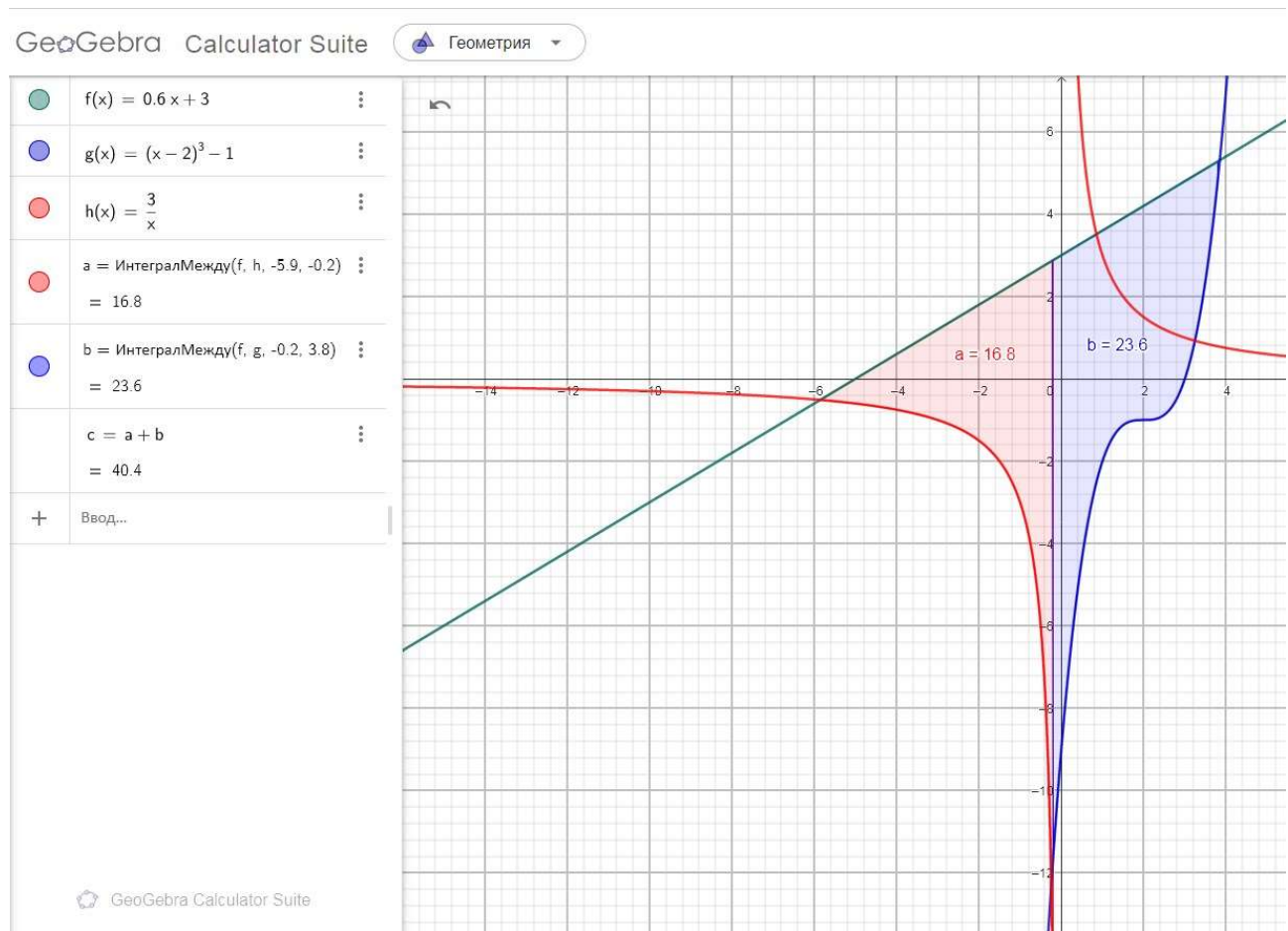


Рис.1 Вычисление площади фигуры

На рисунках рис.2 рис.3 рис.4 показан поиск пересечений функций.

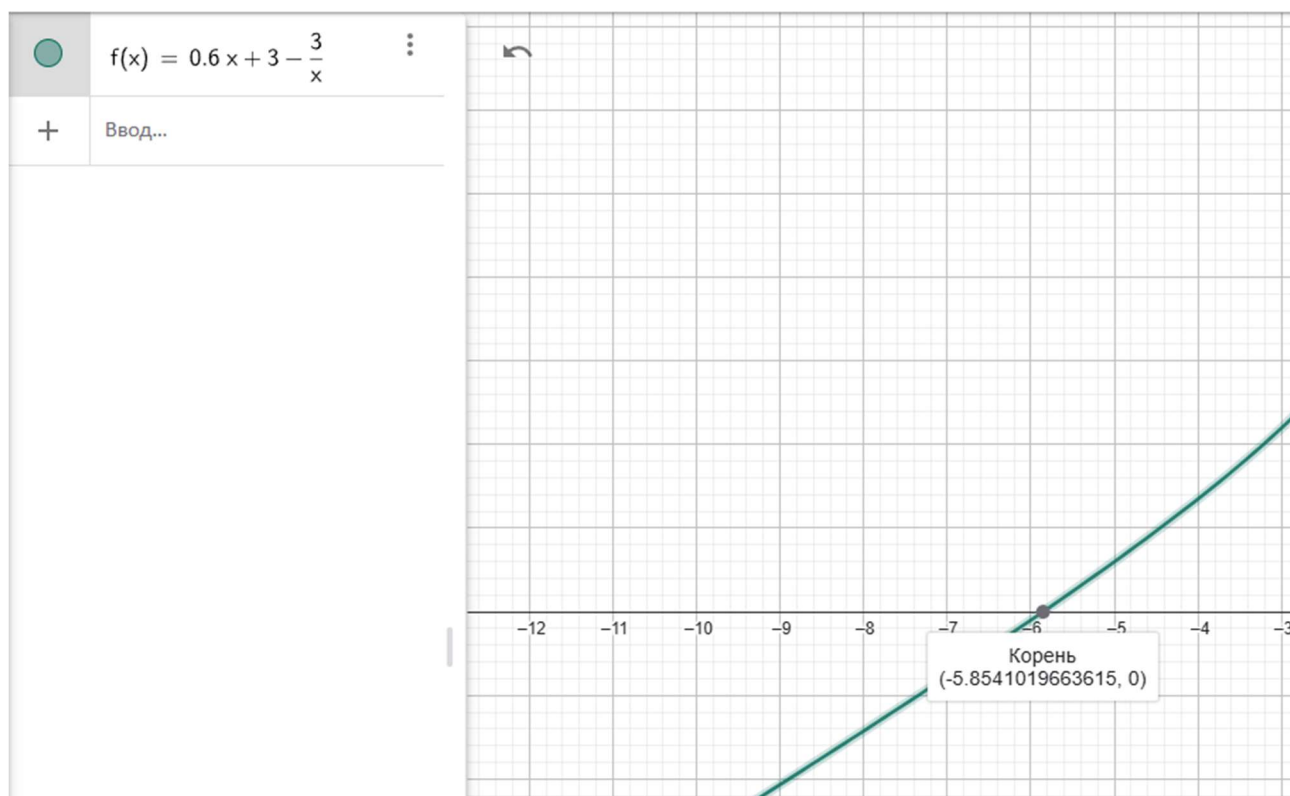


Рис 2. Первый корень

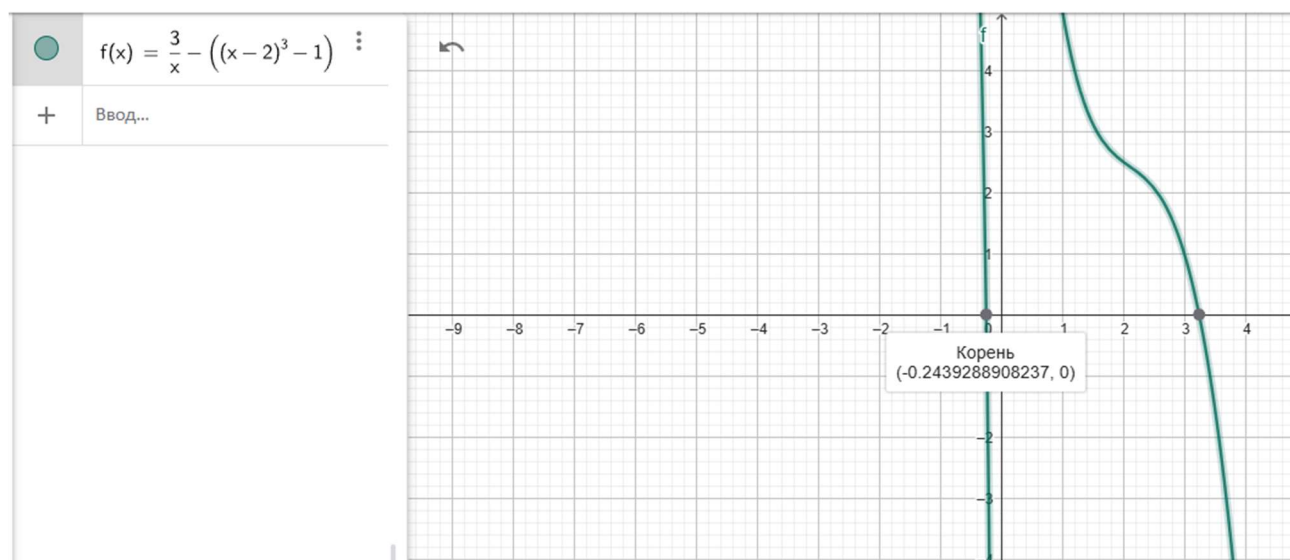


Рис 3. Второй корень

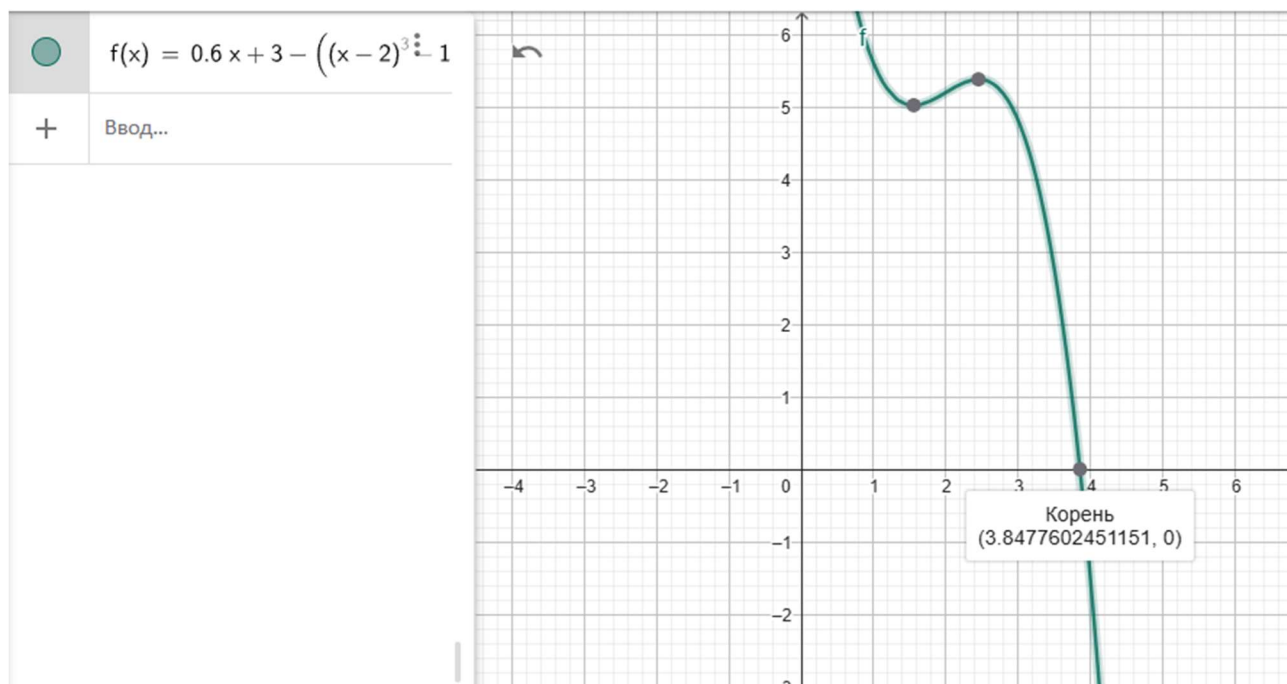


Рис. 4 3й корень

# Решение через SMATH

$$f1(x) := 0,6 \cdot x + 3$$

$$f2(x) := (x - 2)^3 - 1$$

$$f3(x) := \frac{3}{x}$$

$$\text{root1} := \text{solve}(f1(x) - f3(x); x)$$

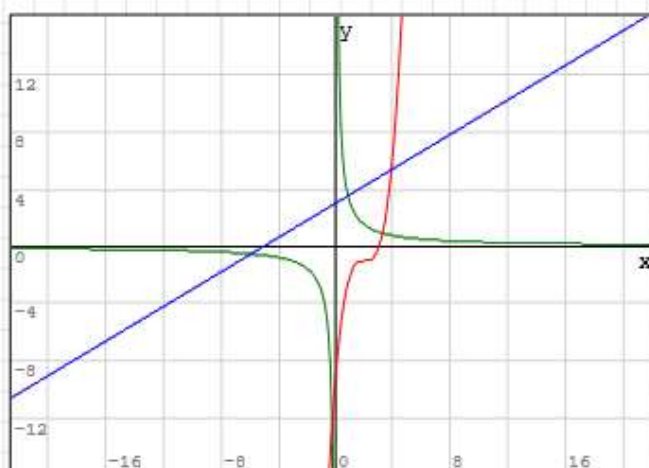
$$\text{root1} = \begin{bmatrix} -5,85410196625 \\ 0,85410196625 \end{bmatrix}$$

$$\text{root1}_1 = -5,85410196625$$

$$\text{root2} := \text{solve}(f3(x) - f2(x); x)$$

$$\text{root2} = \begin{bmatrix} -0,243928853904 \\ 3,243928853904 \end{bmatrix}$$

$$\text{root2}_1 = -0,243928853904$$



$$\begin{cases} f1(x) \\ f2(x) \\ f3(x) \end{cases}$$

$$\text{root3} := \text{solve}(f1(x) - f2(x); x) = 3,847760245115$$

$$Sa := \int_{\text{root1}_1}^{\text{root2}_1} (f1(x) - f3(x)) dx = 16,101297594498$$

$$Sb := \int_{\text{root2}_1}^{\text{root3}} (f1(x) - f2(x)) dx = 24,214614481816$$

$$Sb = \frac{3,89264603528964 \cdot 10^{15}}{160756060692719}$$

$$S := Sa + Sb = 40,315912076314$$

# Исходный код программы

[https://github.com/DmitryLym/DL-repository-0/tree/main/C\\_part2/Coursework2](https://github.com/DmitryLym/DL-repository-0/tree/main/C_part2/Coursework2)

## Описание файлов программы

Программа состоит из следующих файлов:

**main.c** – основной файл программы с точкой входа, функцией main.

**header.h** – файл прототипов функций программы;

**set\_functions.c** – заданные функции, а также первые и вторые производные от них;

**root.c** – содержит функцию поиска корней уравнения комбинированным методом (хорд + касательных) и функции для тестирования;

**integral.c** – содержит функцию расчета интеграла методом трапеций и функции для тестирования;

**interface.c** – файл содержит функцию пользовательского интерфейса;

**makefile** – файл инструкция для утилиты сборки mingw32-make;

**Square.exe** – собранный исполняемый файл программы.



# Описание работы программы

Программа является консольным приложением и рекомендуется запускать её из командной строки.

**Для удобства ниже в каждом пункте работы программы предлагаются соответствующие bat файлы с нужными ключами для удобного запуска демонстрации.**

При запуске без указания ключей, программа кратко выдает предложении ввести ключ «-h» для получения инструкций. И просит указать файл для считывания данных.

Программа допускает применение следующих ключей:

- **«-h»** - получение информации о возможных ключах запуска с кратким описанием их назначения;
- **«-x»** - печать точек пересечения функций;
- **«-c»** - печать количества шагов затраченных комбинированным методом для нахождения корня.
- **«-e»** - задание точности для поиска корня уравнения,  $0 < [e] < 1$ .
- **«-n»** - точность для расчета интеграла,  $[n] > 1$ .
- **«-r»** - запуск тестирования функции поиска корней уравнения
- **«-i»** - запуск тестирования функции вычисления интеграла

## Примеры написания командной строки для запуска:

- **Square.exe -h** – вызов справки
- **Square.exe -r -i** - запуск только тестирования
- **Square.exe -e 0.0001 -n 10000** минимальный набор параметров для запуска программы
- **Square.exe -h -x -c -e 0.0001 -n 10000 -r -i** – максимальное кол-во параметров

# Демонстрация работы программы

## 1. Запуск программы с вызовом помощи

```
G:\Инженер Умных Систем\Учебные Материалы\05. Продвинутый СИ\Лек
found argument "h = (null)".

===== HELP =====
[-h] help
[-x] Print points of intersection
[-c] Print number of iterations
[-e] Calculating with precision [e] the root x of the equation
[-n] Calculating with precision [n] the value of integral
[-r] Test root function
[-i] Test integral function

Press "Enter" for exit.
```

Рис.5.

Файл для запуска **001 key -h.bat**.

## 2. Запуск программы с ключом -r -i

```
G:\Инженер Умных Систем\Учебные Материалы\05. Продвинутый
found argument "r = (null)".
found argument "i = (null)".

-----Test root's function-----
Verification by function  $y=x*x$  and  $y=x$ 
section  $x= [0.5;1.5]$   $\text{eps} = 0.0001$  valid root = 1
-----Root #0-----
Find <<combined method search>> root for 4 steps
Root 0 = 1.000000

-----Test Integral's function-----
Verification by function  $y=1$  and  $y=0$ 
section  $x= [0;100]$   $\text{eps} = 200$  valid value = 100
-----Integral #0-----
S0 = 100.000000

Press "Enter" for exit.
```

Рис.6.

Файл для запуска **002 key -r -i.bat**.

3. Запуск программы с минимальным количеством параметров

Ключи -e 0.0001 -n 10000

```
G:\Инженер Умных Систем\Учебные Материалы\05. Продви
found argument "e = 0.0001".
found argument "n = 10000".

-----Integral #1-----
S1  = 16.093498
-----Integral #2-----
S2  = 24.215851

-----TOTAL SQUARE-----
S   = 40.309349

Press "Enter" for exit.
```

Рис.7.

Файл для запуска **003 key -e -n.bat**

4. Запуск программы с максимальным количеством параметров

Ключи -h -x -c -e 0.0001 -n 10000 -r -i

```
G:\Инженер Умных Систем\Учебные Материалы\05. Продвинутой СИ\Лек
д>Square.exe -h -x -c -e 0.0001 -n 10000 -r -i
found argument "h = (null)".

===== HELP =====
[-h] help
[-x] Print points of intersection
[-c] Print number of iterations
[-e] Calculating with precision [e] the root x of the equation
[-n] Calculating with precision [n] the value of integral
[-r] Test root function
[-i] Test integral function
found argument "x = (null)".
found argument "c = (null)".
found argument "e = 0.0001".
found argument "n = 10000".
found argument "r = (null)".
found argument "i = (null)".

-----Test root's function-----
Verification by function y=x*x and y=x
section x= [0.5;1.5] eps = 0.0001 valid root = 1
-----Root #0-----
Find <<combined method search>> root for 4 steps
Root 0 = 1.000000

-----Test Integral's function-----
Verification by function y=1 and y=0
section x= [0;100] eps = 200 valid value = 100
-----Integral #0-----
S0 = 100.000000

Find <<combined method search>> root for 7 steps
-----Root #1-----
Root 1 = -5.854100
Find <<combined method search>> root for 4 steps
-----Root #2-----
Root 2 = -0.243925
Find <<combined method search>> root for 3 steps
-----Root #3-----
Root 3 = 3.847760

-----Integral #1-----
S1 = 16.093498
-----Integral #2-----
S2 = 24.215851

-----TOTAL SQUARE-----
S = 40.309349
Press "Enter" for exit.
```

Рис.8

Файл для запуска **004 key all.bat**

## Сборка программы утилитой make

```
G:\Инженер Умных Систем\Учебные Материалы\05. Продвинутый СИ\Лекция 8\Курсач 2\Код>mingw32-make
gcc -c -o main.o main.c
gcc -c -o set_functions.o set_functions.c
gcc -c -o root.o root.c
gcc -c -o integral.o integral.c
gcc -c -o Interface.o Interface.c
gcc -o Square main.o set_functions.o root.o integral.o Interface.o
```

Рис.9.

Прим.: файл R.bat – для пересборки проекта.