



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

**Туранин Семен Павлович**

# Оглавление

Постановка задания .....	3
Описание программы .....	4
Исходный код программы.....	4
Описание файлов программы .....	4
Описание работы программы.....	5
Сборка программы утилитой make .....	9

# Постановка задания

Составьте программу, которая находит площадь, образуемую данными уравнениями:

1.  $f_1 = 0,6x + 3$

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

3.  $f_3 = \frac{3}{x}$

## Требования к программе:

1. Основная программа должна поддерживать опции командной строки, при задании которых:
  - Печатаются абсциссы точек пересечения кривых
  - 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  $\varepsilon_1$  корня  $x$  уравнения  $f(x) = g(x)$  на отрезке  $[a, b]$  должно быть реализовано в отдельной Си-функции `root(f, g, a, b, esp1)`. 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  $\varepsilon_2$  величины определённого интеграла от функции  $f(x)$  на отрезке  $[a, b]$  должно быть реализовано в отдельной Си-функции `root(f, a, b, esp2)`
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`

## Требования к аргументам командной строки

Приложение должно обрабатывать аргументы командной строки:

минимальный набор поддерживаемых ключей:

- -h Описание функционала приложения. Список ключей, которые обрабатывает данное приложение и их назначение.
- -t Проверка функций вычисления интеграла (метод прямоугольников и метод трапеций) на тестовой функции (квадрат)
- -p xx - Вывод точек пересечения функций и числа итераций для выбранного метода, где xx - метод вычисления:
  - 1 - метод линейного приближения;
  - 2 - метод хорд;
  - 3 - метод касательных;
  - 4 - комбинированный метод.
- -i Вывод вычисленного интеграла, ограниченного функциями  $f(x) = 0.6x + 3$ ,  $f(x) = (x - 2)^3 - 1$ ,  $f(x) = \frac{3}{x}$ . Интеграл вычисляется методом трапеции и методом прямоугольников
- если нет параметров, то выдается help

## Требования к сборке приложения

- Приложение должно собираться при помощи утилиты make.
- 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

## Исходный код программы

[https://github.com/semenes32/BaseC\\_homework/blob/main/Advanced\\_C/course\\_work/course\\_work.c](https://github.com/semenes32/BaseC_homework/blob/main/Advanced_C/course_work/course_work.c)

## Описание файлов программы

Программа состоит из следующих файлов:

Корневая папка:

makefile – файл инструкция для утилиты сборки mingw32-make;

prog.exe – собранный исполняемый файл программы;

curse.c – основной файл программы с точкой входа, функцией main. Содержит только логику запуска функций и обработку ключей, поступающих от командной строки;

Папка info:

картинка с графиком функций

## Описание работы программы

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

При запуске без указания ключей, программа кратко выведет информацию о своем назначении и предложении ввести ключ «-h» для получения инструкций.

Программа допускает применение следующих ключей:

- «-h» - описание функционала приложения. Список ключей, которые обрабатывает данное приложение и их назначение.
- «-t» - проверка функций вычисления интеграла (метод прямоугольников и метод трапеций) на тестовой функции (квадрат)
- «-p xx» - вывод точек пересечения функций и числа итераций для выбранного метода, где xx - метод вычисления:
  - 1 - метод линейного приближения;
  - 2 - метод хорд;
  - 3 - метод касательных;
  - 4 - комбинированный метод.
- «-i» - вывод вычисленного интеграла, ограниченного функциями  $f(x) = 0.6x + 3$ ,  $f(x) = (x - 2)^3 - 1$ ,  $f(x) = \frac{3}{x}$ . Интеграл вычисляется методом трапеции и методом прямоугольников.

### Примеры написания командной строки для запуска:

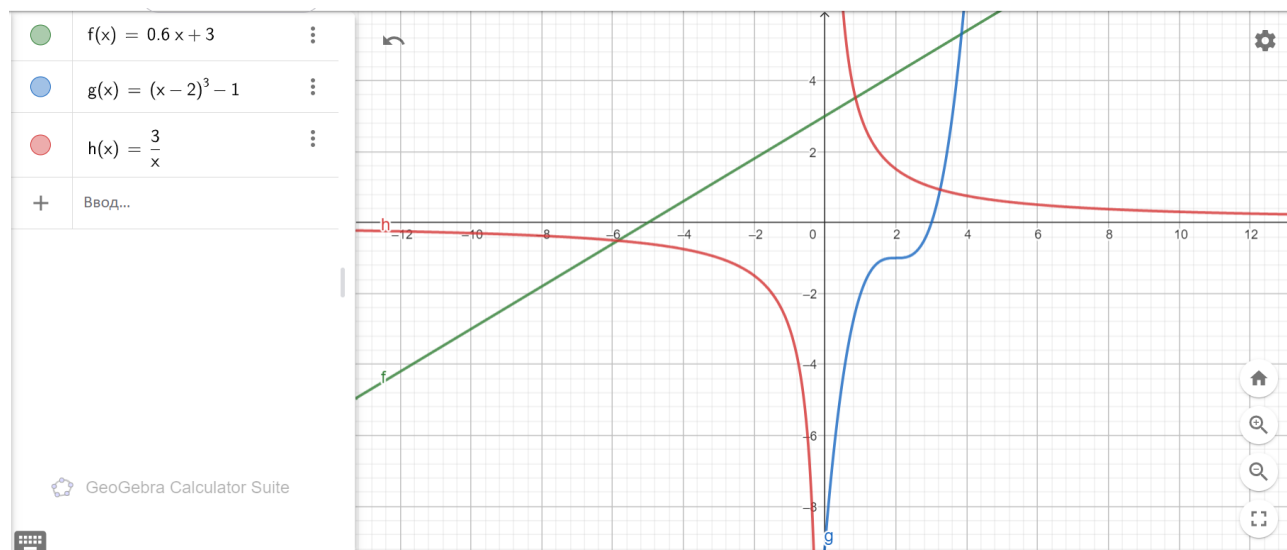
.\prog -h

.\prog -t

.\prog -p 3

.\prog -i

## Графики функций



## Демонстрация работы программы

### Образцы снимков:

Рисунок 1. Запуск программы с опцией «-h»

```
PS C:\Users\79084\Desktop\homework\Advanced_C\course_work> ./prog -h
Usage: C:\Users\79084\Desktop\homework\Advanced_C\course_work\prog.exe [-h] [-p <method number>] [-i] [-t]
Options:
  -h          Display this help message
  -t          Test calcIntegralSquare and calcIntegralTrap functions
  -p          Display points between functions  $f(x)=0.6x+3$ ,  $f(x)=(x-2)^3-1$  and  $f(x)=3/x$ 
  -i          Display CalcIntegral between functions  $f(x)=0.6x+3$ ,  $f(x)=(x-2)^3-1$  and  $f(x)=3/x$ 
```

Рисунок 2. Запуск программы с опцией «-t»

```
ПРОБЛЕМЫ  ВЫХОДНЫЕ ДАННЫЕ  КОНСОЛЬ ОТЛАДКИ  ТЕРМИНАЛ  ПОРТЫ  ESP-IDF  + v prog [icon] [icon] ... ^ x

PS C:\Users\79084\Desktop\homework\Advanced_C\course_work> ./prog -t
-----
Test calcIntegralSquare and calcIntegralTrap functions
calcIntegralSquare integral 4.000000
calcIntegralTrap = 3.995963
```

Рисунок 3. Запуск программы с опцией «-p» и опцией выбора метода вычисления – метод линейного приближения

```
ПРОБЛЕМЫ  ВЫХОДНЫЕ ДАННЫЕ  КОНСОЛЬ ОТЛАДКИ  ТЕРМИНАЛ  ПОРТЫ  ESP-IDF  + v prog [icon] [icon] ... ^ x

PS C:\Users\79084\Desktop\homework\Advanced_C\course_work> ./prog -p 1
Find point between  $f(x)=0.6x+3$  and  $f(x)=(x-2)^3-1$ 
Find Line Search root for 1001 steps
Line Search root = 3.847975
-----
Find point between  $f(x)=0.6x+3$  and  $f(x)=3/x$ 
Find Line Search root for 1000 steps
Line Search root = 0.854017
-----
Find point between  $f(x)=(x-2)^3-1$  and  $f(x)=3/x$ 
Find Line Search root for 1001 steps
Line Search root = 3.243982
```

Рисунок 4. Запуск программы с опцией «-p» и опцией выбора метода вычисления – метод хорд

```
ПРОБЛЕМЫ  ВЫХОДНЫЕ ДАННЫЕ  КОНСОЛЬ ОТЛАДКИ  ТЕРМИНАЛ  ПОРТЫ  ESP-IDF  + v prog [icon] [icon] ... ^ x

PS C:\Users\79084\Desktop\homework\Advanced_C\course_work> ./prog -p 2
Find point between  $f(x)=0.6x+3$  and  $f(x)=(x-2)^3-1$ 
Find Chord Search root for 2 steps
Find Chord Search root = 3.847760
-----
Find point between  $f(x)=0.6x+3$  and  $f(x)=3/x$ 
Find Chord Search root for 2 steps
Find Chord Search root = 0.854106
-----
Find point between  $f(x)=(x-2)^3-1$  and  $f(x)=3/x$ 
Find Chord Search root for 2 steps
Find Chord Search root = 3.243927
```

Рисунок 5. Запуск программы с опцией «-p» и опцией выбора метода вычисления – метод касательных

```

ПРОБЛЕМЫ  ВЫХОДНЫЕ ДАННЫЕ  КОНСОЛЬ ОТЛАДКИ  ТЕРМИНАЛ  ПОРТЫ  ESP-IDF  + v prog [icon] [icon] ... ^ x

PS C:\Users\79084\Desktop\homework\Advanced_C\course_work> ./prog -p 3
Find point between f(x)=0.6x+3 and f(x)=(x-2)^3-1
Find Tangent Search root for 3 steps
Find Tangent root = 3.847760
-----
Find point between f(x)=0.6x+3 and f(x)=3/x
Find Tangent Search root for 3 steps
Find Tangent root = 0.854102
-----
Find point between f(x)=(x-2)^3-1 and f(x)=3/x
Find Tangent Search root for 3 steps
Find Tangent root = 3.243929

```

Рисунок 6. Запуск программы с опцией «-p» и опцией выбора метода вычисления – комбинированный метод

```

ПРОБЛЕМЫ  ВЫХОДНЫЕ ДАННЫЕ  КОНСОЛЬ ОТЛАДКИ  ТЕРМИНАЛ  ПОРТЫ  ESP-IDF  + v prog [icon] [icon] ... ^ x

PS C:\Users\79084\Desktop\homework\Advanced_C\course_work> ./prog -p 4
Find point between f(x)=0.6x+3 and f(x)=(x-2)^3-1
Find Combine Search root for 2 steps
Find Combine root = 3.847688
-----
Find point between f(x)=0.6x+3 and f(x)=3/x
Find Combine Search root for 3 steps
Find Combine root = 0.854102
-----
Find point between f(x)=(x-2)^3-1 and f(x)=3/x
Find Combine Search root for 3 steps
Find Combine root = 3.243929

```

Рисунок 7. Запуск программы с опцией «-i»

```

ПРОБЛЕМЫ  ВЫХОДНЫЕ ДАННЫЕ  КОНСОЛЬ ОТЛАДКИ  ТЕРМИНАЛ  ПОРТЫ  ESP-IDF  + v prog [icon] [icon] ... ^ x

PS C:\Users\79084\Desktop\homework\Advanced_C\course_work> ./prog -i
-----CalcIntegralSquare method-----
calcIntegralSquare under function f(x)=0.6x+3 between point [0.854017,3.8477] = 13.200974
calcIntegralSquare under function f(x)=3/x between point [0.854017,3.243929] = 4.006866
calcIntegralSquare under function f(x)=(x-2)^3-1 between point [3.243929,3.8477] = 1.709932
calcIntegralSquare total integral =7.484176
-----
-----CalcIntegralTrap method-----
calcIntegralTrap under function f(x)=0.6x+3 between point [0.854017,3.8477] = 13.193142
calcIntegralTrap under function f(x)=3/x between point [0.854017,3.243929] = 3.995389
calcIntegralTrap under function f(x)=(x-2)^3-1 between point [3.243929,3.8477] = 1.710695
calcIntegralTrap total integral =7.487059


```



## Сборка программы утилитой make

*Прикрепите в окошке ниже снимок сборки программы утилитой make.*

Рисунок 4. Сборка программы утилитой make



```
ПРОБЛЕМЫ  ВЫХОДНЫЕ ДАННЫЕ  КОНСОЛЬ ОТЛАДКИ  ТЕРМИНАЛ  ПОРТЫ  ESP-IDF  + v powershell [icon] [icon] ... ^ x

PS C:\Users\79084\Desktop\homework\Advanced_C\course_work> gcc -c -o curse_work.o curse_work.c
PS C:\Users\79084\Desktop\homework\Advanced_C\course_work> mingw32-make
gcc -o prog curse_work.o
PS C:\Users\79084\Desktop\homework\Advanced_C\course_work> |
```