

# ***Лабораторная работа №1***

по дисциплине «Программирование на Си»

## **Основные конструкции языка**

Кострицкий А. С., Ломовской И. В.

Москва — 2021 — TS2102271806

### **Содержание**

<b>1</b>	<b>Цель работы</b>	<b>1</b>
<b>2</b>	<b>Первая часть</b>	<b>2</b>
2.1	Задача №1 . . . . .	2
2.2	Задача №2 . . . . .	3
2.3	Задача №3 . . . . .	4
2.4	Задача №4 . . . . .	4
<b>3</b>	<b>Вторая часть</b>	<b>4</b>
3.1	Задача №5 . . . . .	5
3.2	Задача №6 . . . . .	5
3.3	Задача №7 . . . . .	6
3.4	Задача №8 . . . . .	6
3.5	Задача №9 . . . . .	7
<b>4</b>	<b>Взаимодействие с системой тестирования</b>	<b>7</b>

## **1 Цель работы**

Целью лабораторной работы является знакомство студента с инструментарием для создания, отладки и сопровождения программ, написанных на языке программирования Си. Студент должен получить и закрепить на практике следующие знания и умения:

1. Умение компилировать программу из командной строки в рамках однофайлового проекта.
2. Умение работать в интегрированной среде разработки на примере Qt Creator, в том числе
  - (a) создавать проект;
  - (b) настраивать различные варианты сборки и очистки проекта;
  - (c) отлаживать программу.
3. Умение декомпозировать задачи на подзадачи.

4. Умение реализовывать подзадачи в виде «чистых» функций.
5. Умение продумывать ограничения на входные данные и процесс обработки ошибочных ситуаций.
6. Умение подготавливать тестовые данные и оценивать полноту оных.

## 2 Первая часть

Входные данные вводятся, а выходные выводятся строго в том порядке, в котором они встречаются в задании. Вариант для каждой задачи определяется отдельно, по умолчанию по кольцу (как остаток от деления варианта студента в ЭУ на количество вариантов), но может быть изменён преподавателем на его усмотрение.

Помните, что кодируется не задача, а её решение, которое желательно иметь *до* того, как Вы сели за клавиатуру. Попробуйте поступить следующим образом:

1. Выделите в задаче входные и выходные данные. Обычно это те данные, которые предлагается ввести самостоятельно и требуется распечатать соответственно.
2. Найдите однозначную зависимость между входными и выходными данными, иначе говоря, решите задачу аналитически.
3. Выберите подходящие типы для представления данных.
4. Мысленно разделите решение на три блока — блок ввода, блок расчёта, блок вывода; в первом закодируйте ввод и, если необходимо, проверку на валидность всех входных данных; во втором запишите последовательность действий для вычисления выходных данных по входным; в третьем реализуйте распечатку выходных данных на экран.

Для задач первой части можно считать, что пользователь всегда вводит только валидные данные, никаких проверок ввода совершать не нужно.

### 2.1 Задача №1

Результат выполнения каждого пункта задачи необходимо отразить в отчёте. Для написания отчёта используется Wiki репозитория студента на кафедральном GitLab. Если на момент выполнения лабораторной работы проект в GitLab ещё не создан, отчёт можно оформить в виде текстового файла, информацию из которого после создания репозитория нужно будет перенести в Wiki.

1. Установите дома необходимое программное обеспечение. Если в процессе установки или настройки программного обеспечения вы столкнулись с какими-либо проблемами, опишите в отчёте проблемы и способы, с помощью которых Вы эти проблемы преодолели.
2. Познакомьтесь с сообщениями, которые выдаёт компилятор во время компиляции программы с синтаксическими ошибками:
  - (a) Попытайтесь скомпилировать программу из задачи №1, которая заведомо содержит синтаксические ошибки.

- (b) Изучите выдачу компилятора. Какая информация помогает вам понять, к какой строке исходной программы относится сообщение об ошибке?
  - (c) Исправьте ошибки. Ошибки исправляются строго по одной и по порядку обнаружения с последующей попыткой компиляции после каждого исправления. В отчёт выносятся «тройки»: сообщение компилятора об ошибке, Ваша интерпретация этого сообщения и соответствующие исправления. Можно посоветовать следующую последовательность действий: компиляция; копирование сообщения о первой ошибке в отчёт; написание в отчёте ее объяснения; исправление ошибки; использование команды `git diff`, чтобы показать суть исправлений; использование команды `git commit` для фиксации изменений.
3. Познакомьтесь с интегрированной средой разработки на примере QT Creator. В результате знакомства Вы должны уметь:
- (a) Создавать проект, настраивать этапы сборки и очистки проекта, запуск приложения.
  - (b) Анализировать сообщения об ошибках компиляции; знать, где посмотреть сообщения об ошибках, выданные самим компилятором.
  - (c) Использовать отладчик: выполнять программу в пошаговом режиме; устанавливать точки останова и условия их срабатывания; работать с переменными.

Найдите и исправьте ошибки в программе.

```
include studio.h
main{
(
int s;

s: = 56;
print (Year has s weeks)
)
```

## 2.2 Задача №2

### Варианты

0. Принять с клавиатуры величины оснований  $a$  и  $b$  и высоты  $h$  равнобедренной трапеции. Найти и вывести на экран периметр  $P$  трапеции.
1. Принять с клавиатуры величины оснований  $a$  и  $b$  и угла в градусах  $\varphi$  при большем основании равнобедренной трапеции. Найти и вывести на экран площадь  $S$  трапеции.
2. Принять с клавиатуры координаты  $x_a, y_a, \dots, y_c$ , треугольника  $abc$  на плоскости. Найти и вывести на экран периметр  $P$  треугольника.
3. Принять с клавиатуры величины двух сторон  $a$  и  $b$  треугольника  $abc$  и угла в градусах  $\varphi$  между ними. Найти и вывести на экран площадь  $S$  треугольника.

## 2.3 Задача №3

### Варианты

0. По введённым с клавиатуры росту человека в сантиметрах, обхвату грудной клетки в сантиметрах и массе тела в килограммах определить и вывести на экран нормальный вес человека  $m_{\text{normal}}$  и индекс массы его тела BMI, полагая, что

$$m_{\text{normal}} = \frac{ht}{240},$$

$$\text{BMI} = \frac{m}{h^2},$$

где  $h$  — рост человека, измеряемый в сантиметрах при расчёте нормального веса, и в метрах — при расчёте индекса массы тела;  $t$  — обхват грудной клетки в сантиметрах;  $m$  — вес в килограммах.

1. С клавиатуры вводится число литров  $V_1$  и температура  $T_1$  воды в первой ёмкости и число литров  $V_2$  и температура  $T_2$  воды во второй ёмкости, которые предстоит смешать. Найти и вывести на экран объём  $V$  и температуру  $T$  образовавшейся смеси.
2. С клавиатуры вводятся величины сопротивлений трёх резисторов,  $R_1$ ,  $R_2$ ,  $R_3$ , соединённых параллельно. Найти и вывести на экран сопротивление  $R$  всего соединения.
3. С клавиатуры вводится величины начальной скорости  $v_0$ , ускорения  $a$  и времени  $t$  равноускоренного прямолинейного движения тела. Найти и вывести на экран перемещение  $S$ .

## 2.4 Задача №4

### Варианты

0. Бутылка воды стоит 45 копеек. Пустые бутылки сдаются по 20 копеек, и на полученные деньги опять покупается вода. По введённому с клавиатуры количеству копеек найти и вывести на экран наибольшее возможное количество бутылок воды, которые можно купить. Циклов не использовать.
1. Определить и вывести на экран номер подъезда и этажа по введённому с клавиатуры номеру квартиры девятиэтажного дома, считая, что на каждом этаже ровно 4 квартиры, а нумерация квартир начинается с первого подъезда.
2. С клавиатуры задаётся время в секундах. Перевести в часы, минуты, секунды и вывести на экран.
3. С клавиатуры задаётся трёхзначное число. Вывести на экран произведение его цифр.

## 3 Вторая часть

Входные данные вводятся, а выходные выводятся строго в том порядке, в котором они встречаются в задании. Вариант для каждой задачи определяется отдельно, по умолчанию по кольцу (как остаток от деления варианта студента в ЭУ на количество вариантов), но может быть изменён преподавателем на его усмотрение.

Начиная с пятой задачи, следует проверять валидность входных данных. Перечень и примеры проверок Вы можете найти в методических указаниях в кафедральном moodle.

Начиная с пятой задачи, следует отслеживать полноту тестовых данных с помощью утилиты gsov. Подробные методические указания Вы можете найти в кафедральном moodle.

Для реализации каждой из задач второй части Вам необходимо выделить, по крайней мере, одну осмысленную функцию.

### 3.1 Задача №5

#### Варианты

0. С клавиатуры вводятся целое  $a$  и целое положительное  $n$ . Вычислить и вывести на экран  $a^n$ .
1. С клавиатуры вводятся целое положительное  $a$  и целое положительное  $d$ . Вычислить и вывести на экран частное  $q$  и остаток  $r$  при делении  $a$  на  $d$ , не используя операций целочисленного деления.
2. Вычислить и вывести на экран число Фибоначчи  $\text{Fib}_n$ , приняв с клавиатуры целое неотрицательное  $n$ . Рекурсивных функций не использовать, положить  $\text{Fib}_0 = 0$ ,  $\text{Fib}_1 = 1$ . Предусмотреть возможное переполнение целого числа при решении.
3. Приняв с клавиатуры два натуральных числа,  $a$  и  $b$ , вычислить и вывести на экран наибольший общий делитель оных.
4. Составить программу для печати разложения на простые множители заданного натурального числа  $n$ . Если  $n$  равно 1, печатать ничего не надо.
5. Составить программу, распечатывающую вводимое с клавиатуры натуральное  $n$ , полагая, что функцию `printf` можно вызывать только для печати цифр числа: вызов `printf("%d", i)` можно осуществлять лишь при  $i = 0, 1, 2, \dots, 9$ .

### 3.2 Задача №6

#### Варианты

0. Принять с клавиатуры координаты  $x_a, y_a, \dots, y_c$  треугольника  $abc$  на плоскости. Определить тип треугольника и вывести на экран целое число в зависимости от ответа: 0 — остроугольный, 1 — прямоугольный, 2 — тупоугольный.
1. Принять с клавиатуры координаты  $x_a, y_a, \dots, y_c$  треугольника  $abc$  на плоскости. Найти и вывести на экран площадь  $S$  треугольника.
2. С клавиатуры задаются координаты  $x_a, y_a, \dots, y_c$  вершин треугольника  $abc$  на плоскости и координаты точки  $x_p, y_p$ . Определить, лежит ли точка внутри, на границе или вне треугольника, и вывести на экран 0, 1 или 2 соответственно.
3. С клавиатуры задаются координаты точек  $x_q, y_q, x_r, y_r$  на прямой и координаты произвольной точки  $x_p, y_p$ . Определить взаимное расположение точки и прямой и вывести на экран целое число в зависимости от ответа: 0 — лежит выше прямой, 1 — на прямой, 2 — под прямой.
4. С клавиатуры задаются координаты точек  $x_q, y_q, x_r, y_r$  отрезка  $qr$  и координаты произвольной точки  $x_p, y_p$ . Определить, не принадлежит или принадлежит точка отрезку, и вывести на экран 0 или 1 соответственно.

5. С клавиатуры задаются координаты точек  $x_p, y_p, x_q, y_q$  отрезка  $pq$  и координаты точек  $x_r, y_r, x_s, y_s$  отрезка  $rs$ . Определить, не пересекаются или пересекаются два отрезка, и вывести на экран 0 или 1 соответственно.

### 3.3 Задача №7

Приняв с клавиатуры  $x$  и  $\varepsilon$ ,  $0 < \varepsilon \leq 1$ , вычислить и вывести на экран вычисленное с точностью<sup>1</sup>  $\varepsilon$  приближённое значение  $s(x)$  и точное значение  $f(x)$  функции  $f$ , абсолютную  $\Delta$  и относительную  $\delta$  погрешности:

$$\Delta = |f(x) - s(x)|, \quad \delta = \frac{|f(x) - s(x)|}{|f(x)|}.$$

#### Варианты

0.

$$\forall x \, f(x) = e^x, \, s(x) = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \frac{x^5}{5!} + \dots$$

1.

$$\forall x \, f(x) = \sin(x), \, s(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} + \dots$$

2.

$$(\forall x : |x| \leq 1) \, f(x) = \arcsin(x), \, s(x) = x + \frac{x^3}{2 \cdot 3} + \frac{3 \cdot x^5}{2 \cdot 4 \cdot 5} + \frac{3 \cdot 5 \cdot x^7}{2 \cdot 4 \cdot 6 \cdot 7} + \dots$$

3.

$$(\forall x : |x| \leq 1) \, f(x) = \arctan(x), \, s(x) = x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \frac{x^9}{9} + \dots$$

4.

$$(\forall x : |x| < 1) \, f(x) = \frac{1}{(1+x)^3}, \, s(x) = 1 - \frac{2 \cdot 3 \cdot x}{2} + \frac{3 \cdot 4 \cdot x^2}{2} - \frac{4 \cdot 5 \cdot x^3}{2} + \frac{5 \cdot 6 \cdot x^4}{2} + \dots$$

5.

$$(\forall x : |x| < 1) \, f(x) = \frac{1}{\sqrt{1-x^2}}, \, s(x) = 1 + \frac{1 \cdot x^2}{2} + \frac{1 \cdot 3 \cdot x^4}{2 \cdot 4} + \frac{1 \cdot 3 \cdot 5 \cdot x^6}{2 \cdot 4 \cdot 6} + \frac{1 \cdot 3 \cdot 5 \cdot 7 \cdot x^8}{2 \cdot 4 \cdot 6 \cdot 8} + \dots$$

### 3.4 Задача №8

Если во время выполнения программы в этой задаче ошибок не было, вывод результата нужно начинать с сообщения «**Result:**□», за которым и выводится результат вычислений. В случае возникновения ошибочной ситуации вывод следует начинать с сообщения «**Error:**□», за которым должно следовать описание ошибочной ситуации.

---

<sup>1</sup>Накопление суммы следует выполнять до тех пор, пока очередной член ряда по абсолютной величине не окажется меньше заданной  $\varepsilon$ .

## Варианты

0. Вводятся беззнаковое целое число длиной 4 байта  $a$  и целое число  $n$ . Числа вводятся в десятичной системе счисления. Необходимо циклически сдвинуть значение переменной  $a$  на  $n$  позиций вправо. Результат вывести в двоичной системе счисления.
1. Вводятся значения 4-х байт в десятичной системе счисления. Необходимо реализовать «упаковку» этих байт в беззнаковое целое (первый байт занимает самые старшие биты беззнакового числа, последний – самые младшие) и «распаковку» обратно. Вывести сначала результат «упаковки» (беззнаковое целое) в двоичной системе счисления, затем значения четырёх байт, полученных в результате «распаковки» (от старшего к младшему) в десятичной системе счисления.
2. Вводятся беззнаковое целое число длиной 4 байта  $a$  и целое число  $n$ . Числа вводятся в десятичной системе счисления. Необходимо циклически сдвинуть значение переменной  $a$  на  $n$  позиций влево. Результат вывести в двоичной системе счисления.
3. Вводится беззнаковое целое число длиной 4 байта. Число вводится в десятичной системе счисления. Зашифровать значение числа, поменяв местами чётные и нечётные биты (т.е. местами меняются 0-ой и 1-ый биты, 2-ой и 3-ий, и т.д.). Результат шифрования вывести на экран в двоичной системе счисления.

### 3.5 Задача №9

Приняв с клавиатуры по концевому признаку (до первого отрицательного числа) элементы непустой последовательности неотрицательных чисел  $x$ , вычислить и вывести на экран значение  $g$ .

## Варианты

0.

$$g = \sqrt{x_1 + \frac{x_2}{2} + \dots + \frac{x_n}{n}}.$$

1.

$$g = \sin \left( \sqrt{x_1} + \sqrt{\frac{x_2}{2}} + \dots + \sqrt{\frac{x_n}{n}} \right).$$

2.

$$g = \frac{1}{n} (\sqrt{1+x_1} + \sqrt{2+x_2} + \dots + \sqrt{n+x_n}).$$

3.

$$g = \exp \left( \frac{1}{(x_1+1) \cdot (x_2+2) \cdot \dots \cdot (x_n+n)} \right).$$

## 4 Взаимодействие с системой тестирования

1. Исходный код лабораторной работы размещается студентом в ветви `lab_LL`, а решение каждой из задач — в отдельной папке с названием вида `lab_LL_PP_CC`, где `LL` — номер лабораторной, `PP` — номер задачи, `CC` — вариант студента. Если дана общая задача без вариантов, решение следует сохранять в папке с названием вида `lab_LL_PP`.

Пример: решения восьми задач седьмого варианта пятой лабораторной размещаются в папках `lab_05_01_07`, `lab_05_02_07`, `lab_05_03_07`, ..., `lab_05_08_07`.

2. Исходный код должен соответствовать оглашённым в начале семестра правилам оформления.
3. Если для решения задачи студентом создаётся отдельный проект в IDE, разрешается поместить под версионный контроль файлы проекта, добавив перед этим необходимые маски в список игнорирования. Старайтесь добавлять маски общего вида. Для каждого проекта должны быть созданы, как минимум, два варианта сборки: **Debug** — с отладочной информацией, и **Release** — без отладочной информации.
4. Для каждой программы ещё до реализации студентом заготавливаются и помещаются под версионный контроль в подпапку **func\_tests** функциональные тесты, демонстрирующие её работоспособность.

Позитивные входные данные следует располагать в файлах вида **pos\_TT\_in.txt**, выходные — в файлах вида **pos\_TT\_out.txt**, аргументы командной строки при наличии — в файлах вида **pos\_TT\_args.txt**, где **TT** — номер тестового случая.

Негативные входные данные следует располагать в файлах вида **neg\_TT\_in.txt**, выходные — в файлах вида **neg\_TT\_out.txt**, аргументы командной строки при наличии — в файлах вида **neg\_TT\_args.txt**, где **TT** — номер тестового случая.

Разрешается помещать под версионный контроль в подпапку **func\_tests** сценарии автоматического прогона функциональных тестов. Если Вы используете при автоматическом прогоне функциональных тестов сравнение строк, не забудьте проверить используемые кодировки. Помните, что **UTF-8** и **UTF-8(BOM)** — две разные кодировки.

Под версионный контроль в подпапку **func\_tests** также помещается файл **readme.md** с описанием в свободной форме содержимого каждого из тестов. Вёрстка файла на языке **Markdown** обязательной не является, достаточно обычного текста.

Пример: восемь позитивных и шесть негативных функциональных тестов без дополнительных ключей командной строки должны размещаться в файлах **pos\_01\_in.txt**, **pos\_01\_out.txt**, ..., **neg\_06\_out.txt**. В файле **readme.md** при этом может содержаться следующая информация:



```

# Тесты для лабораторной работы №LL

## Входные данные
Целые a, b, c

## Выходные данные
Целые d, e

## Позитивные тесты:
- 01 - обычный тест;
- 02 - в качестве первого числа ноль;
...
- 08 - все три числа равны.

## Негативные тесты:
- 01 - вместо первого числа идёт буква;
- 02 - вместо второго числа идёт буква;
...
- 06 - вводятся слишком большие числа.

```

5. Если не указано обратное, успешность ввода должна контролироваться. При первом неверном вводе программа должна прекращать работу с ненулевым кодом возврата.
6. Вывод программы может содержать текстовые сообщения и числа. Если не указано обратное, тестовая система анализирует числа в потоке вывода, поэтому они могут быть использованы только для вывода результатов — использовать числа в информационных сообщениях запрещено.

Пример: сообщение «Input point 1:» будет неверно воспринято тестовой системой, а сообщения «Input point A:» или «Input first point:» — правильно.

Тестовая система вычлняет из потока вывода числа, обособленные пробельными символами.

Пример: сообщения «a=1.043» и «a = 1.043.» будут неверно восприняты тестовой системой, а сообщения «a: 1.043» или «a = 1.043» — правильно.

7. Если не указано обратное, числа двойной точности следует выводить, округляя до шестого знака после точки.