

# Программирование

Факультет безопасности информационных технологий  
Университет ИТМО

Осень 2023

## Лабораторная работа № 2 Указатели и динамическая память

Разработать на языке C для ОС Linux программу, которая выполняет заданную операцию над матрицей чисел заданного типа (Табл. 2). Матрицу следует размещать в динамической памяти способом, определяемым вариантом из Табл. 1. Размер матрицы указывает пользователь после запуска программы. Матрица либо заполняется пользователем вручную, либо заполняется случайными числами. Программа выводит исходную матрицу на экран, осуществляет операцию согласно варианту работы (см. ниже), после чего выводит преобразованную матрицу на экран.

Номер варианта лабораторной работы составлен из двух чисел (X-Y), где X — номер варианта из Табл. 1, а Y — номер варианта из Табл. 2.

Матрица — таблица чисел, например, матрица A, показанная ниже, имеет M строк и N столбцов, нумерация строк и столбцов начинается с единицы. Каждый элемент имеет два индекса, первый — номер строки, второй — номер столбца:

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1N} \\ a_{21} & a_{22} & \dots & a_{2N} \\ \dots & \dots & \dots & \dots \\ a_{M1} & a_{M2} & \dots & a_{MN} \end{bmatrix}$$

Программа должна представлять собой консольное приложение, настройка работы которого осуществляется путем передачи аргументов в строке запуска (квадратные скобки показывают, что соответствующие аргументы могут отсутствовать):

```
lab2abcNXXXXX [-m] число_строк число_столбцов
```

Имя программы должно начинаться на lab2, далее должен следовать уникальный для варианта суффикс. Уникальный суффикс составляется из первых букв имени, отчества (если есть) и фамилии студента, выполняющего лабораторную работу. Далее следует номер группы студента. Используются строчные латинские буквы и арабские (в традиционном понимании, т. е. 0..9) цифры. Например, если студента, выполняющего лабораторную, зовут Петр Сергеевич Иванов, его группа — N32451, то имя программы должно быть lab2psiN32451.

Оба аргумента (число\_строк и число\_столбцов) представляют собой целые числа, оба аргумента обязательны. Перед аргументами может быть указана опция -m (квадратные скобки означают необязательность опции), означающая ввод элементов матрицы вручную. Например (тип числа — double, преобразование — замена строк столбцами, наклонный шрифт означает ввод с клавиатуры, ↵ — нажатие на Enter):

```
$ ./lab2psiN32451 -m 3 2
```

Введите строки матрицы:

```
3.14 3.42↵
```

```
5.17 6.18↵
```

```
4.01 0.11↵
```

Исходная матрица:

```
3.14 3.42
```

```
5.17 6.18
```

```
4.01 0.11
Результат:
3.14 5.17 4.01
3.42 6.18 0.11
```

Если опция `-m` отсутствует, элементы матрицы должны быть заполнены случайными числами:

```
$ ./lab2psiN32451 2 2
Исходная матрица:
0.12 0.87
1.33 5.08
Результат:
0.12 1.33
0.87 5.08
```

Если пользователь при запуске указывает неподдерживаемые опции, лишние аргументы, вместо чисел нужного формата строки, которые не являются числами или не могут поместиться в заданный тип, либо совершает какие-то другие ошибки, программа должна сообщить об этом:

```
$ ./lab2psiN32451 ping pong
Ошибка: 'ping' не является числом.

$ ./lab2psiN32451 -k 5 5
Ошибка: опция '-k' не поддерживается.

$ ./lab2psiN32451 -m 5
Ошибка: отсутствует обязательный аргумент.

$ ./lab2psiN32451 -m 3 3
Введите строки матрицы:
3.14 щито 1,11
Ошибка: 'щито' не является числом.
```

В некоторых вариантах требуется задать дополнительные данные для работы программы (например, количество элементов для сдвига  $T$ , порядок кольца  $K$  и так далее). В этом случае их можно либо передавать в качестве дополнительных аргументов командной строки, либо считывать с консоли после запуска программы.

Проект (исходные коды, заголовочные файлы, Makefile и прочие файлы, которые могут понадобиться для сборки) должен содержаться в отдельном каталоге с именем, совпадающим с названием программы (lab2abcNXXXXX), и собираться с помощью стандартной утилиты make. Исходные файлы программы на языке C должны компилироваться с помощью gcc. Makefile должен поддерживать как минимум цели `all` и `clean`. Пример заготовки проекта ЛР № 2 содержится на [гугл-диске](#) в папке "лабораторные" (архив lab2abcNXXXXX.tar.gz, для распаковки можно использовать команду `tar -xzf lab2abcNXXXXX.tar.gz`)

Порядок выполнения и сдачи лабораторной работы:

1. Скачать заготовку проекта, изменить название каталога на правильное (соответствующее вашей группе и ФИО), скорректировать содержимое Makefile'a.
2. Выполнить задание, подготовить все файлы проекта, скомпилировать программу и библиотеку с флагами `-Wall -Wextra -Werror` и устранить все предупреждения и ошибки.
3. Протестировать программу на различных входных данных, убедиться, что ошибок нет, в противном случае вернуться к пункту 2.
4. Удалить все исполняемые и промежуточные файлы из папки проекта (`make clean`). В архиве должны остаться только файлы \*.c, \*.h, Makefile, README.txt.

5. Заархивировать папку проекта, используя формат .tar.gz. (tar -czvf lab2abcNXXXXXX.tar.gz lab2abcNXXXXXX/).
6. Подготовить отчет по лабораторной работе в формате pdf, на титульной странице отчета не забыть поставить подпись. Файл отчета должен иметь название NXXXXXX\_ФамилияИО\_LP2.pdf. Состав отчета описан ниже.
7. Отправить архив и отчет в формате pdf на почту преподавателя, который ведет лабораторные, письмом с темой «Программирование LP2 Фамилия Имя Отчество NXXXXXX Вариант X-Y».
8. Дождаться ответа по почте или на лабораторном занятии, устранить возможные замечания (повторить с пункта 1).
9. Получить некоторое количество вопросов от преподавателя по отчету и темам, связанным с лабораторной, и дать на них ответы (а может и не получить, если лабораторная выполнена на хорошем уровне и сомнений в знаниях студента у преподавателя не возникает). Получить от преподавателя подтверждение, что работа выполнена успешно и отчет принят.
10. Немного расслабиться и приступить к следующей лабораторной работе :-)

Отчет должен быть подготовлен в формате pdf и содержать:

- правильно оформленную титульную страницу (с подписью студента);
- задание;
- Make-файл;
- примеры работы программ на различных исходных данных (скриншоты);
- исходный текст программы с комментариями.

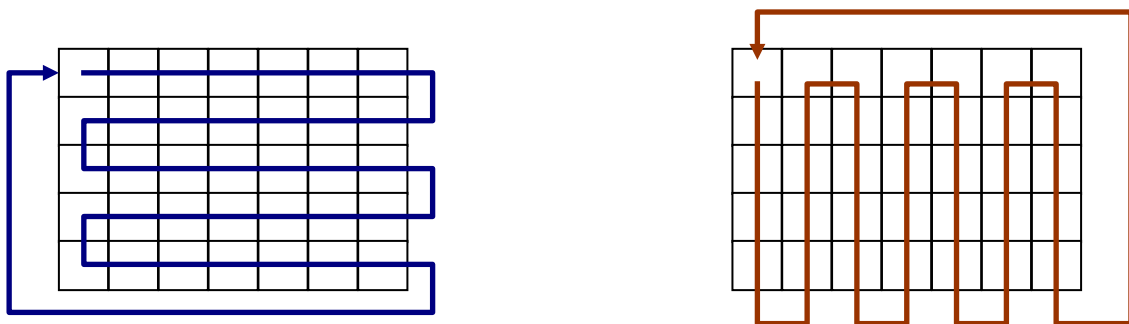


Рисунок 1. Варианты циклического сдвига элементов в матрице

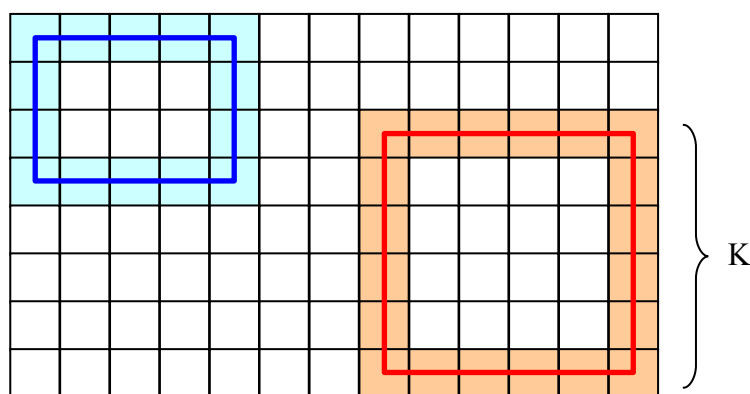
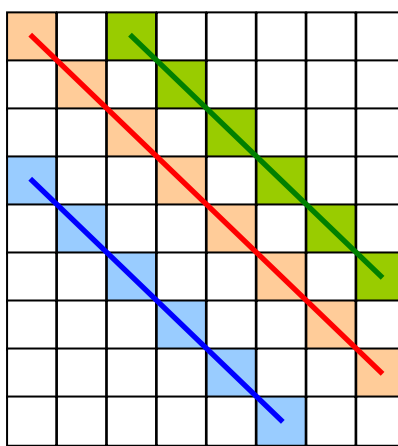


Рисунок 2. Кольцо порядка 6 и  
8x13



цепочка в матрице размером

Рисунок 3. Диагонали порядков 0 (красный цвет), 2 (зеленый) и –3 (синий) в матрице размером 9x8

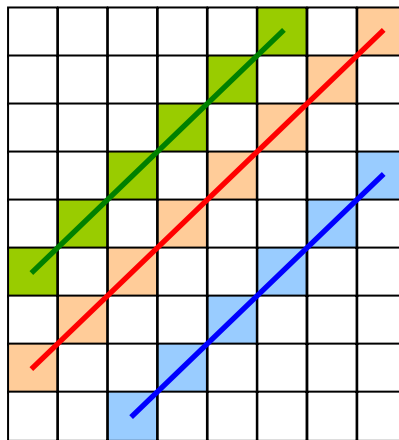


Рисунок 4. Обратные диагонали порядков 0 (красный цвет), 2 (зеленый) и –3 (синий) в матрице размером 9x8

**Замечание 1.** В некоторых вариантах встречаются термины *кольцо*, *цепочка*, *диагональ*, *обратная диагональ*.

*Кольцо* задается позицией углового элемента (например, верхнего левого элемента  $a_{LT}$ ) и количеством элементов  $K$ , образующих сторону квадрата (на Рис. 2 элементы, образующие кольцо порядка 6 выделены красным цветом).

*Цепочка* задается позициями элементов  $a_{LT}$ ,  $a_{RT}$ ,  $a_{RB}$ ,  $a_{LB}$ , где  $a_{LT}$  – левый верхний элемент,  $a_{RT}$  – правый верхний элемент,  $a_{RB}$  – правый нижний элемент и  $a_{LB}$  – левый нижний элемент (на Рис. 2 элементы, образующие цепочку, выделены синим цветом).

*Диагональ* задается порядком  $V$  (сдвигом относительно главной диагонали). На Рис. 3 красным цветом выделены элементы, расположенные на диагонали порядка 0, синим – элементы, расположенные на диагонали порядка –3, зеленым – элементы, расположенные на диагонали порядка 2.

*Обратная диагональ* задается порядком  $V$  (сдвигом относительно главной обратной диагонали). На Рис. 4 красным цветом выделены элементы, расположенные на обратной диагонали порядка 0, синим – элементы, расположенные на обратной диагонали порядка –3, зеленым – элементы, расположенные на обратной диагонали порядка 2.

**Замечание 2.** При выполнении лабораторной работы следует использовать только функции стандартной библиотеки C и системные вызовы операционной системы. Использовать C++, ввод-вывод в стиле C++ (классы `ifstream/ofstream/...`), контейнеры и алгоритмы STL (`<string>`, `<vector>`, `<map>`, ...) **запрещено**.

**Замечание 3.** В программе должна присутствовать обработка ошибок: в случаях, если пользователь передал некорректные аргументы или ввел недопустимые значения, программа должна выдавать диагностическое сообщение на консоль (в стандартный поток ошибок), прежде чем завершиться.

**Замечание 4.** В результате преобразования должна измениться матрица, просто вывести результат преобразования на экран – недостаточно.

**Замечание 5.** Код, выполняющий операцию над матрицей, желательно оформить в виде функции, которой передается матрица (или указатель на матрицу — в зависимости от варианта хранения матрицы в памяти) и размерность матрицы.

**Замечание 6.** Если для облегчения понимания того, как выполняется преобразование, нужно выводить дополнительную информацию, её желательно выводить с помощью переменной окружения LAB2DEBUG (см. примеры на [гугл-диске](#)). В вариантах, где требуется вычисление характеристик столбца, строки и т.д., следует предусмотреть возможность вывода значений вычисленных значений на экран в режиме отладки.

**Замечание 7.** Для преобразования строки в число рекомендуется использовать функцию `sscanf()` с правильно выбранным спецификатором формата и проверкой возвращаемого значения этой функции и переменной `errno`.

**Замечание 8.** Программа должна успешно компилироваться и выполняться в 64-разрядной ОС Linux с ядром версии  $\geq 5.0$ , glibc версии  $\geq 2.0$ , gcc версии  $\geq 10.0$ .

**Таблица 1. Способ представления матрицы**

№ варианта (X)	Способ представления матрицы
1	Одномерный массив, содержащий строки матрицы
2	Одномерный массив, содержащий столбцы матрицы
3	Массив указателей на строки матрицы
4	Массив указателей на столбцы матрицы
5	VLA

**Таблица 2. Тип числа, над которым требуется выполнять преобразование**

№ варианта (Y)	Преобразование и тип элементов
1	<i>Задание:</i> Назовем характеристикой столбца сумму его элементов, расположенных на нечетных позициях (нумерация элементов в матрице начинается с единицы). Расположить столбцы в порядке убывания их характеристик. <i>Тип элементов:</i> int
2	<i>Задание:</i> Осуществить сдвиг кольца порядка K на T элементов по часовой стрелке. Величины K, T и позицию верхнего левого элемента кольца вводит пользователь после запуска программы. <i>Тип элементов:</i> float
3	<i>Задание:</i> Исключить из матрицы столбцы и строки, в которых нечетных элементов меньше, чем четных. <i>Тип элементов:</i> long
4	<i>Задание:</i> Назовем характеристикой столбца корень квадратный из произведения элементов столбца. Расположить столбцы в порядке возрастания их характеристик. <i>Тип элементов:</i> double
5	<i>Задание:</i> Назовем характеристикой строки сумму его четных элементов. Расположить строки в порядке убывания их характеристик. <i>Тип элементов:</i> long long
6	<i>Задание:</i> Назовем характеристикой столбца результат поразрядной операции XOR между элементами столбца. Осуществить циклический сдвиг каждого столбца на количество единиц в шести младших битах значения характеристики. <i>Тип элементов:</i> uint16_t
7	<i>Задание:</i> Осуществить циклический сдвиг на T элементов диагонали порядка V. Величины T и V вводит пользователь после запуска программы. <i>Тип элементов:</i> float
8	<i>Задание:</i> Назовем характеристикой строки или столбца результат поразрядной операции AND между элементами строки или столбца. Если количество единичных битов в значениях характеристик соседних строк или столбцов разное, то вставить между ними нулевую строку или столбец.

	<i>Тип элементов:</i> uint32_t
9	<p><i>Задание:</i> Осуществить сдвиг цепочки, заданной позициями угловых элементов <math>a_{LT}</math>, <math>a_{RT}</math>, <math>a_{RB}</math>, <math>a_{LB}</math>, на Т элементов против часовой стрелки. Позиции угловых элементов и величину сдвига Т пользователь вводит после запуска программы.</p> <p><i>Тип элементов:</i> unsigned long</p>
10	<p><i>Задание:</i> Назовем характеристикой столбца или строки целую часть корня квадратного из суммы элементов столбца или строки. Исключить из матрицы столбцы и строки с одинаковыми характеристиками. В результате в матрице характеристики всех столбцов и строк должны быть уникальными.</p> <p><i>Тип элементов:</i> double</p>
11	<p><i>Задание:</i> Найти в исходной матрице положение матрицы Р размером 2х2. Элементы матрицы Р пользователь вводит с клавиатуры. Все элементы найденной матрицы Р в исходной матрице требуется возвести в квадрат.</p> <p><i>Тип элементов:</i> short</p>
12	<p><i>Задание:</i> Осуществить в матрице циклический сдвиг на Т элементов так, как показано на Рис. 1а.</p> <p><i>Тип элементов:</i> long double</p>
13	<p><i>Задание:</i> Назовем характеристикой столбца сумму его элементов, кратных числу В. Расположить столбцы матрицы в порядке убывания характеристик столбцов. Число В пользователь вводит после запуска программы.</p> <p><i>Тип элементов:</i> long double</p>
14	<p><i>Задание:</i> Определить все строки и столбцы матрицы, в которых встречаются одинаковые элементы.</p> <p><i>Тип элементов:</i> long</p>
15	<p><i>Задание:</i> Осуществить в матрице циклический сдвиг на Т элементов так, как показано на Рис. 1б.</p> <p><i>Тип элементов:</i> float</p>
16	<p><i>Задание:</i> Назовем характеристикой столбца сумму его элементов, расположенных на четных позициях (нумерация элементов матрицы начинается с единицы). Расположить столбцы в порядке возрастания их характеристик.</p> <p><i>Тип элементов:</i> uint64_t</p>
17	<p><i>Задание:</i> Назовем характеристикой строки результат поразрядной операции «стрелка Пирса» между элементами строки. Осуществить циклический сдвиг каждой строки на количество нулей в младшем байте значения характеристики.</p> <p><i>Тип элементов:</i> uint16_t</p>
18	<p><i>Задание:</i> Назовем характеристикой кольца сумму его элементов. Найти в исходной матрице положение кольца порядка К с максимальной характеристикой.</p> <p><i>Тип элементов:</i> short</p>
19	<p><i>Задание:</i> Осуществить в матрице циклический сдвиг на Т элементов влево каждой диагонали с нечетным порядком и на Т элементов вправо каждой диагонали с четным порядком.</p> <p><i>Тип элементов:</i> double</p>
20	<p><i>Задание:</i> В квадратной матрице NxN осуществить сдвиг на Т элементов внешнего кольца порядка N против часовой стрелки, вложенного в него кольца порядка N-1 – на Т элементов по часовой стрелке, вложенного в него кольца порядка N-2 – на Т элементов против часовой стрелки и т.д.</p> <p><i>Тип элементов:</i> long</p>
21	<p><i>Задание:</i> Назовем характеристикой столбца остаток от деления суммы элементов столбца на количество элементов в нем. Сдвинуть циклически каждый столбец матрицы на значение его характеристики. Четные столбцы сдвигать влево, нечетные – вправо.</p> <p><i>Тип элементов:</i> unsigned int</p>
22	<p><i>Задание:</i> Назовем характеристикой кольца сумму его элементов. Сдвинуть два первых найденных кольца порядка К с одинаковыми характеристиками на V элементов по часовой стрелке.</p> <p><i>Тип элементов:</i> short</p>

23	<p><i>Задание:</i> Назовем характеристикой кольца произведение его элементов. Найти в исходной матрице положение кольца порядка K с минимальной характеристикой и сдвинуть его против часовой стрелки на L элементов.</p> <p><i>Тип элементов:</i> double</p>
24	<p><i>Задание:</i> Назовем характеристикой столбца корень квадратный из суммы квадратов его четных элементов. Расположить столбцы в порядке возрастания их характеристик.</p> <p><i>Тип элементов:</i> int16_t</p>
25	<p><i>Задание:</i> Осуществить сдвиг кольца порядка K на S элементов против часовой стрелки. Величины K, S и позицию нижнего правого элемента кольца вводит пользователь после запуска программы.</p> <p><i>Тип элементов:</i> short</p>
26	<p><i>Задание:</i> Назовем характеристикой строки корень квадратный из суммы элементов строки. Расположить строки в порядке убывания их характеристик.</p> <p><i>Тип элементов:</i> float</p>
27	<p><i>Задание:</i> Назовем характеристикой строки сумму её элементов, старшие тетрады которых четные. Расположить строки в порядке возрастания их характеристик.</p> <p><i>Тип элементов:</i> short</p>
28	<p><i>Задание:</i> Назовем характеристикой строки результат поразрядной операции "штрих Шеффера" между элементами строки. Осуществить циклический сдвиг каждой строки на количество единиц в пяти старших битах значения характеристики.</p> <p><i>Тип элементов:</i> uint32_t</p>
29	<p><i>Задание:</i> Осуществить циклический сдвиг на T элементов обратной диагонали порядка V. Величины T и V вводит пользователь после запуска программы.</p> <p><i>Тип элементов:</i> int32_t</p>
30	<p><i>Задание:</i> Найти в исходной матрице положение подматрицы S размером 3x3, сумма элементов которой равна X. Число X пользователь вводит с клавиатуры. Все элементы найденной матрицы S в исходной матрице нужно обнулить.</p> <p><i>Тип элементов:</i> int</p>
31	<p><i>Задание:</i> Осуществить в матрице циклический сдвиг на L элементов в направлении, обратном тому, которое показано на Рис. 1а.</p> <p><i>Тип элементов:</i> unsigned int</p>
32	<p><i>Задание:</i> Назовем характеристикой строки сумму её элементов, имеющих нечетное значение. Расположить строки в порядке возрастания их характеристик.</p> <p><i>Тип элементов:</i> int</p>
33	<p><i>Задание:</i> Назовем характеристикой столбца результат поразрядной операции «стрелка Пирса» между элементами столбца. Осуществить циклический сдвиг каждого столбца на количество единиц в старшем байте значения характеристики.</p> <p><i>Тип элементов:</i> uint32_t</p>
34	<p><i>Задание:</i> Назовем характеристикой кольца сумму квадратов его элементов. Найти в исходной матрице положение кольца порядка K с минимальной характеристикой.</p> <p><i>Тип элементов:</i> float</p>
35	<p><i>Задание:</i> Осуществить в матрице циклический сдвиг на L элементов влево каждой обратной диагонали с нечетным порядком и на R элементов вправо каждой обратной диагонали с четным порядком.</p> <p><i>Тип элементов:</i> unsigned long</p>
36	<p><i>Задание:</i> Назовем характеристикой строки целую часть результата деления квадрата количества элементов строки на корень квадратный из суммы элементов строки. Сдвинуть циклически каждую строку матрицы на значение её характеристики. Четные строки сдвигать право, нечетные – влево.</p> <p><i>Тип элементов:</i> float</p>
37	<p><i>Задание:</i> Осуществить в матрице циклический сдвиг на V элементов в направлении, обратном тому, которое показано на Рис. 1б.</p> <p><i>Тип элементов:</i> double</p>

38	<p><i>Задание:</i> Назовем характеристикой строки сумму дробных частей её элементов. Расположить строки в порядке возрастания их характеристик.</p> <p><i>Тип элементов:</i> float</p>
39	<p><i>Задание:</i> Осуществить в матрице циклический сдвиг на V элементов в направлении, обратном тому, которое показано на Рис. 1б., каждого второго элемента.</p> <p><i>Тип элементов:</i> int</p>
40	<p><i>Задание:</i> Осуществить в матрице циклический сдвиг на T элементов так, как показано на Рис. 1а., каждого второго элемента.</p> <p><i>Тип элементов:</i> long double</p>

Версия 0.4 от 20.11.2023

Список изменений:

01.11.2023 v0.3: Исправлена опечатка в Табл. 2, вар. 26: было "... из суммы элементов столбца".

20.11.2023 v04.: Добавлено описание способа передачи дополнительных аргументов.