

КАЗАНСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ

ИНСТИТУТ ФИЗИКИ

*Кафедра вычислительной физики
и моделирования физических процессов*

И. И. Файрушин, Р. М. Хуснутдинов, А. В. Сазанов, А. В. Мокшин

СБОРНИК ЗАДАНИЙ ПО КУРСУ «ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ»

Учебно-методическое пособие

Казань – 2021

УДК 53.072; 530.1; 530.145

ББК 22.311.4

X 98

*Принято на заседании кафедры вычислительной физики и моделирования
физических процессов*

Протокол № 7 от 1 июля 2021 года

Рецензенты:

кандидат физико-математических наук,
доцент кафедры вычислительной физики КФУ

Б.Н. Галимзянов;

кандидат технических наук,
доцент кафедры общей физики КНИТУ-КАИ

Т.Я. Асадуллин

Файрушин И.И., Хуснутдинов Р.М., Сазанов А.В., Мокшин А. В.

Сборник заданий по курсу «Информационные технологии»: учебно-методическое пособие / И.И. Файрушин, Р.М. Хуснутдинов, А.В. Сазанов, А.В. Мокшин. – Казань: Изд-во Казан. ун-та, 2021. – 40 с.

ISBN

В данном учебно-методическом пособии представлены задания по курсу «Информационные технологии», а также примеры с решениями задач по программированию на языке C/C++. Настоящее пособие предназначено для студентов физических специальностей высших учебных заведений при изучении дисциплин, связанных с программированием, информационными технологиями, численными методами и компьютерным моделированием. В сборнике представлены типовые задания и темы проектных работ.

УДК 53.072; 530.1; 530.145

ББК 22.311.4

ISBN

©И.И. Файрушин, Р.М. Хуснутдинов,
А.В. Сазанов, А.В. Мокшин, 2021

Содержание

Предисловие	4
Установка редактора исходного кода Code::Blocks	5
1 Массивы	8
1.1 Одномерные массивы	8
1.2 Сортировка и упорядочение массивов	8
2 Файлы	10
3 Структуры	11
4 Обработка строк	14
5 Динамические типы данных	16
6 Рекурсия	17
7 Функции и указатели	18
7.1 Указатели	18
7.2 Подпрограммы-функции	19
7.3 Функции и указатели	20
7.4 Функции и файлы	21
8 Объектно-ориентированное программирование	23
8.1 Классы	23
8.2 Шаблоны функций	26
8.3 Векторы	26
8.4 Алгоритмы STL	27
9 Общие задачи	29
Ответы и указания	32
Темы проектных работ	39
Список рекомендуемой литературы	40

Предисловие

В данном учебно-методическом пособии представлены задания по курсу «Информационные технологии», а также примеры с решениями задач по программированию на языке C/C++. Предлагаемое пособие предназначено для организации самостоятельной и аудиторной работы на практических занятиях по курсам «Информационные технологии» и «Программирование», а также на лабораторных занятиях по учебной практике «Практика по получению первичных профессиональных умений и навыков» со студентами Института физики КФУ. Часть заданий сопровождается подробными ответами и указаниями, что облегчает самостоятельную работу студентов. Основное содержание пособия составляют задания средней трудности, но имеются и задачи повышенной сложности. В конце пособия приводится список тем проектных работ, которые студенты должны выполнять самостоятельно. Учебное пособие предназначено для студентов физических специальностей высших учебных заведений. Пособие составлено с целью повышения эффективности организации самостоятельной работы и аудиторных занятий студентов очного и заочного отделений физического и физико-математического факультетов. Кроме того, пособие может быть рекомендовано для использования при проведении факультативных занятий студентов на других естественнонаучных факультетах.

Установка редактора исходного кода Code::Blocks

Для написания собственных программ на языке программирования C++ необходимо установить любую из возможных сред разработки. Например, свободную кроссплатформенную среду разработки — **Code::Blocks**.

- Скачиваем дистрибутив с сайта codeblocks.org.
- После установки на рабочем столе появляется иконка с одноименным названием, или же в меню **Пуск**→**Все программы**.
- В открывшемся окне выбираем **Create a new project** (рис. 1):

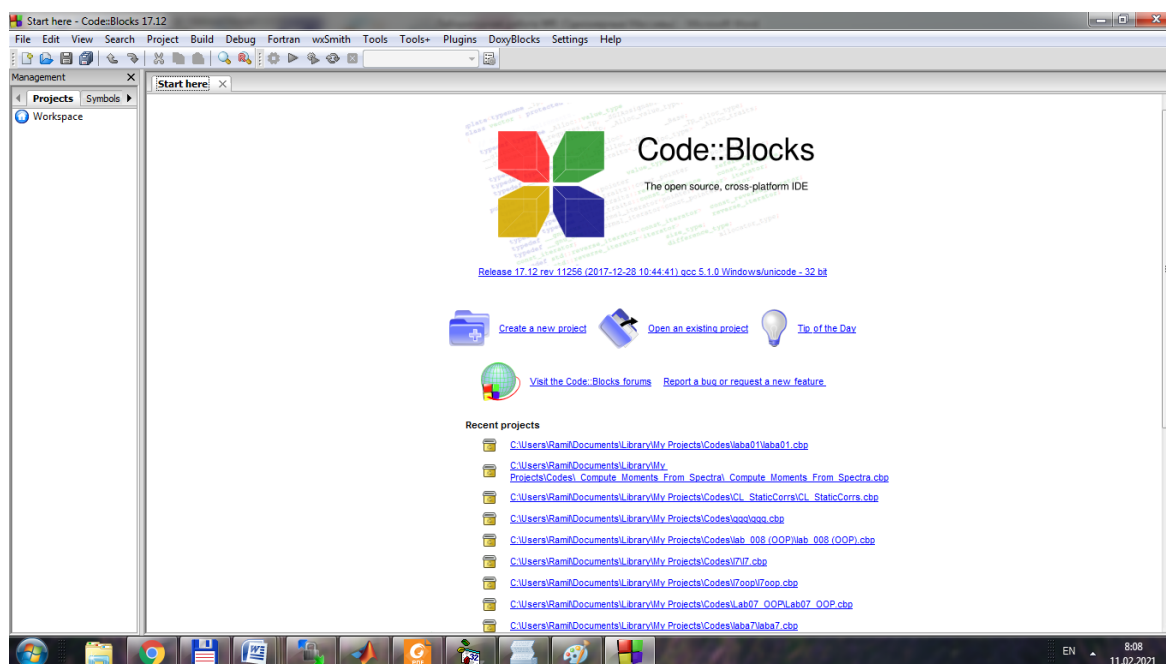


Рис. 1: Запуск программы Code::Blocks

- Выбираем иконку **Console Application** (рис. 2):

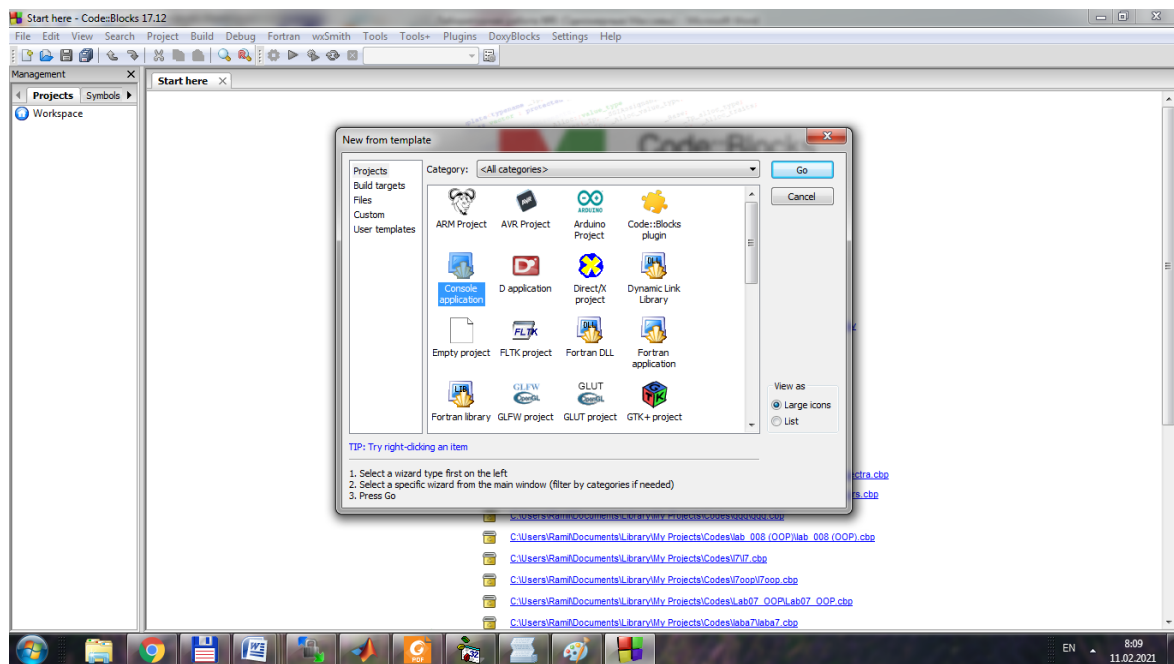


Рис. 2: Окно выбора иконки Console Application

- Выбираем **C++** (рис. 3):

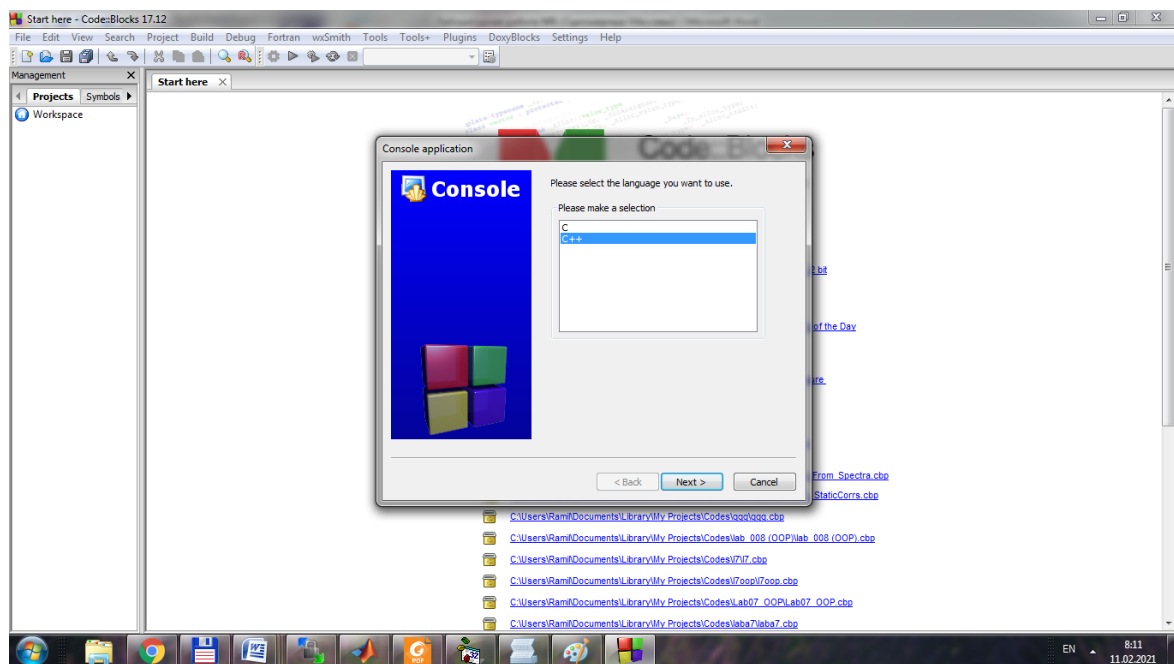


Рис. 3: Выбор языка C++

- Пишем имя проекта, например, **__lab01** (рис. 4):

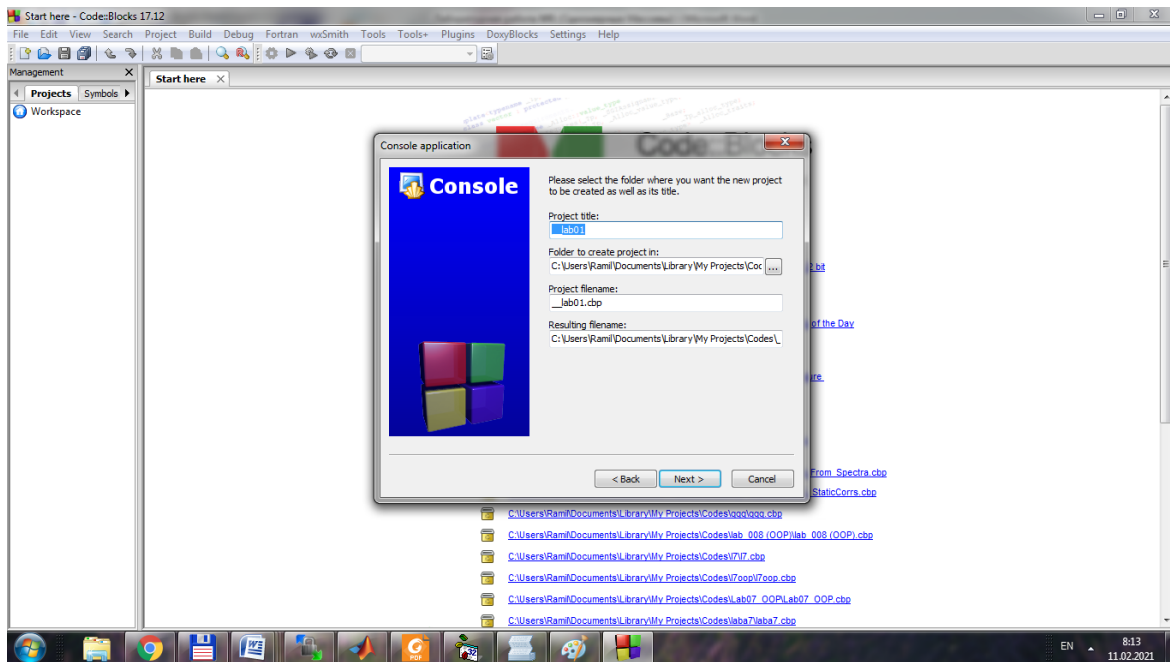


Рис. 4: Ввод имени проекта

- Далее, соглашаемся на все предложенные варианты и кликаем на вкладку **Sources** → **main.cpp**. В результате, получаем следующее окно и шаблон программы **Hello, world!** (рис. 5):

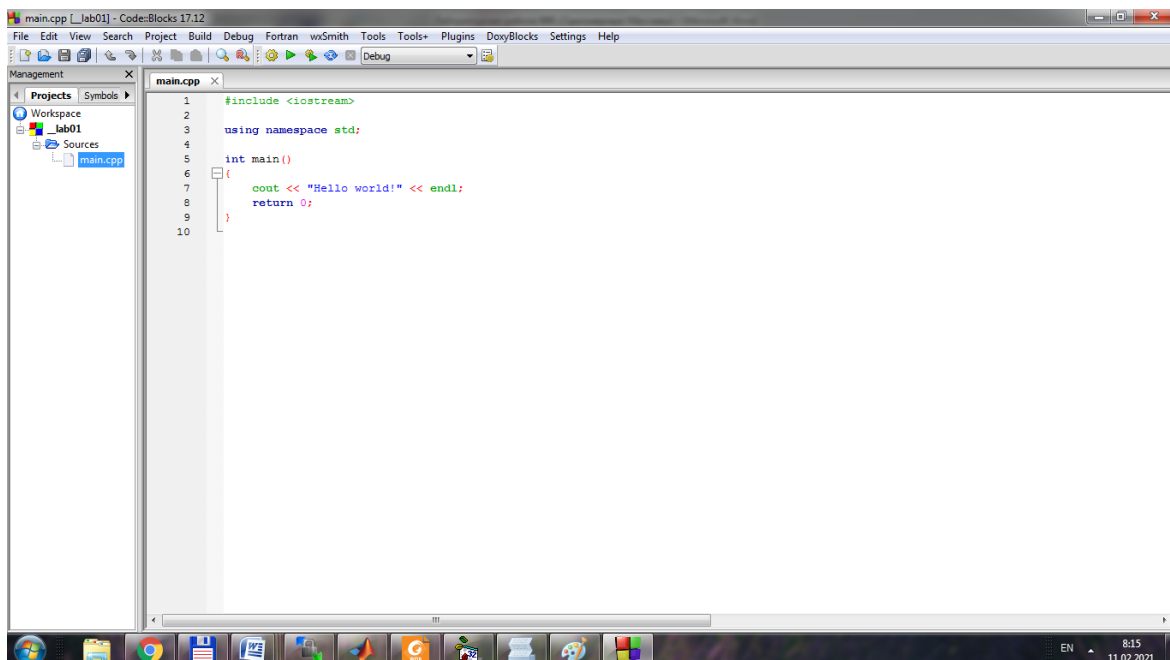


Рис. 5: Программа «Hello, world!» в окне редактора исходного кода

- Для компиляции и запуска программы нажимаем на клавишу **F9 (Build and run)**.

§ 1 Массивы

§ 1.1 Одномерные массивы

1. Дан массив вещественных чисел. Найдите сумму отрицательных элементов массива.
2. Найдите произведение элементов массива с нечетными номерами.
3. Дан массив целых чисел. Количество чисел запросить с клавиатуры. Найти максимальный (минимальный) элемент массива и его номер, при условии, что все элементы различные.
4. Найдите наименьший четный элемент массива. Если такого нет, то выведите первый элемент.
5. Преобразовать массив так, чтобы сначала шли нулевые элементы, а затем все остальные.
6. Ввести массив, в котором только два одинаковых элемента. Определить их местоположение.
7. Ввести два массива действительных чисел. Определить максимальные элементы в каждом массиве и поменять их местами.
8. Задан целочисленный массив. Определить процентное содержание элементов, превышающих среднеарифметическое всех элементов массива.
9. Выполнить сортировку массива по возрастанию (убыванию).
10. Дан массив из 10 элементов. Первые 4 упорядочить по возрастанию, последние 4 по убыванию.

§ 1.2 Сортировка и упорядочение массивов

1. Создайте матрицу случайных чисел размерности $n \times m$ в диапазоне $[1; 10]$.
2. Дана квадратная матрица. Вывести на экран элементы, стоящие на диагонали.
3. Дана матрица. Вывести на экран все нечетные столбцы, у которых первый элемент меньше последнего.

4. Дана матрица $N \times M$ случайных чисел. Отсортировать элементы главной диагонали матрицы по убыванию.
5. Дана матрица $N \times M$ случайных чисел. Упорядочить первый столбец матрицы по возрастанию, а последний столбец — по убыванию.
6. Дан массив из 10 элементов. Отсортируйте отдельно элементы от 0-го по 2-й, с 3-го по 5-й и с 6-го по 9.
7. Дан трехмерный массив $N \times M \times K$ случайных чисел ($N, M, K > 5$). Отсортируйте матрицу $N \times M$ при $K = 2$ и выведите её на экран монитора.
8. Дан массив 20 целых чисел на отрезке $[-2; 5]$. Упорядочить массив, удалив нули со сдвигом влево.
9. Дан массив 20 целых чисел на отрезке $[-5; 5]$. Упорядочить массив, удалив повторяющиеся элементы.
10. Дан массив. Найдите два соседних элемента, сумма которых минимальна.
11. В данном массиве найдите количество чисел, соседи у которых отличаются более чем в 2 раза.
12. Дана матрица. Вывести на экран все четные строки.
13. Найдите сумму номеров минимального и максимального элементов массива.
14. Введите одномерный целочисленный массив. Найдите наибольший нечетный элемент. Далее осуществите циклический сдвиг влево элементов, стоящих справа от найденного максимума.
15. Дан массив размером $n \times n$, элементы которого целые числа. Для каждого столбца подсчитать сумму отрицательных элементов и записать данные в текстовый файл.
16. В двумерном массиве, элементы которого целые числа, удалить все столбцы, в которых первый элемент больше последнего. Результат записать в файл.

§ 2 Файлы

1. Создайте матрицу **$x[n][n]$** случайных чисел. Сохраните все элементы матрицы в файл с названием **Matrix.txt**. Считайте содержимое файла **Matrix.txt** в новый массив **$y[n][n]$** и выведите его на экран дисплея.
2. Напишите программу, которая считывала бы элементы главной диагонали матрицы из файла **Matrix.txt**.
3. Напишите программу, которая удаляла бы k -столбец ($1 < k < M$) в файле **Matrix.txt**.
4. Напишите программу, которая считывала бы элементы матрицы из файла **Matrix.txt** и записывала бы их в массив соответствующего размера. Отсортируйте все столбцы матрицы по убыванию. Полученный массив запишите в файл **Matrix_Sort.txt**.
5. Дан текстовый файл, содержащий целые числа. Удалить из него все четные числа.
6. В данном текстовом файле удалить все слова, которые содержат хотя бы одну цифру.
7. Напишите программу, которая считывала бы саму себя и выводила бы на экран дисплея исходный текст программы в обратном порядке.
8. Имеется файл с текстом. Осуществить шифрование данного текста в новый файл. Осуществить расшифровку полученного текста.

§ 3 Структуры

1. Описать структуру с именем **AEROFLOT**, содержащую следующие поля:
 - название пункта назначения рейса;
 - номер рейса;
 - тип самолета.
2. Написать программу, выполняющую следующие действия:
 - ввод с клавиатуры данных в массив, состоящий из семи элементов типа **AEROFLOT**; записи должны быть размещены в алфавитном порядке по названиям пунктов назначения (для этого выполните процедуру сортировки);
 - вывод на экран пунктов назначения и номеров рейсов, обслуживаемых самолетом, тип которого введен с клавиатуры. Если таких рейсов нет, выдать на дисплей соответствующее сообщение.
3. Описать структуру с именем **STUDENT**, содержащую следующие поля:
 - **NAME** – фамилия и инициалы;
 - **GROUP** – номер группы;
 - **SES** - успеваемость (массив из пяти элементов).
4. Написать программу, выполняющую следующие действия:
 - ввод с клавиатуры данных в массив **STUD1**, состоящий из десяти структур типа **STUDENT**; записи должны быть упорядочены по возрастанию содержимого поля **GROUP**;
 - вывод на дисплей фамилий и номеров групп для всех студентов, включенных в массив, если средний балл студента больше 4,0. Если таких нет, вывести соответствующее сообщение.
5. Информация об итогах сдачи сессии каждым студентом представлена в следующем порядке: Фамилия Имя Отчество, номер группы, экзаменационные оценки по четырем предметам. Отсортируйте фамилии студентов по алфавиту. Определить процент студентов, сдавших экзамены на 4 и 5.

6. Ведомость успеваемости студентов курса содержит следующую информацию: номер группы, фамилию, средний балл за последнюю сессию. Составить список студентов в порядке возрастания их номеров групп.
7. Даны два отсчета времени в часах, минутах и секундах. Найти величину временного интервала в секундах. Код реализовать через составной тип данных.
8. Дано пять различных дат в виде: число, месяц, год. Вывести их на экран в порядке возрастания.
9. Создать массив структур для учета занятости аудитории: день недели, время учебной пары, аудитория, название предмета. Реализовать поиск периодов времени, когда выбранная аудитория свободна.
10. Список книг содержит следующую информацию: фамилии авторов, название книги, год издания. Найти все книги, в названии которых имеется определенное слово, например, "физика".
11. Список имеющихся в продаже автомобилей содержит следующие сведения: марка автомобиля, цвет, стоимость, мощность двигателя, расход бензина на 100 км. Вывести перечень автомобилей, удовлетворяющих определенным требованиям клиента, таким, например, как стоимость в диапазоне 300-500 тыс.руб., расход бензина в пределах 8-10 л и т.п.
12. Описать два комплексных числа и проделать над ними операции сложения, вычитания, умножения и деления.
13. Даны стоимости двух товаров в рублях и копейках. Найти суммарную стоимость покупки и рассчитать сдачу. Квитанцию о покупке (чек) записать в текстовый файл.
14. Ведомость содержит следующие сведения о сдавших вступительные экзамены: ФИО, оценки (баллы) по отдельным дисциплинам, например:

Name	Mathematics	Physics	Informatics
Sidorov R.V.	90	74	58

Вывести на экран фамилии абитуриентов, имеющих средний балл 60 и выше, и их количество.

15. Дано пять различных дат в виде: число, месяц, год. Вывести их на экран в порядке возрастания. Результаты записать в текстовый файл.

16. В расписании рейсов вылетов самолетов на определенный день содержится следующая информация: номер рейса, тип самолета, пункт назначения, время вылета, например:

Fly	Airplane	Destination	Departure
U124	Airbus 90	London	13:46

Определить, какие самолеты и когда летят до заданного пункта назначения. Запишите в текстовый файл исходные данные и результаты выборки.

17. Описать структуру с именем WORKER, содержащую следующие поля:

- NAME — фамилия и инициалы работника;
- POS — название занимаемой должности;
- YEAR — год поступления на работу.

18. Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив TABL, состоящий из десяти структур типа WORKER; записи должны быть размещены по алфавиту.
- вывод на дисплей фамилий работников, чей стаж работы в организации превышает значение, введенное с клавиатуры;
- если таких работников нет, вывести на дисплей соответствующее сообщение.

§ 4 Обработка строк

1. В заданном тексте заменить все символы «+» на « - ». В данной задаче воспользуйтесь массивом символов (Заголовочный файл **cstring**).
2. В данном тексте посчитать число символов «+» и «-».
3. Напишите программу, которая вычисляет длину введенной с клавиатуры строки. Реализуйте код программы, используя строковый тип данных (Заголовочный файл **string**).
4. Задана строка символов. Определить, есть ли заданный символ «э» в этой строке символов. Выведите на экран номер первого вхождения данного символа в строке..
5. Пусть задан некоторый текст. Вычислить, сколько раз повторяется наперед заданный символ «а».
6. Задана некоторая строка символов. Создать новую строку, которая образована из данной строки чтением от конца до начала.
7. Задано слово. Проверить, читается ли это слово слева направо и наоборот. *Простейшие слова-палиндромы*: мим, дед, наган, заказ, кабак, казак, мадам, шалаш.
8. Вводится строка слов, разделенных пробелами. Найти самое длинное слово и вывести его на экран.
9. Пусть имеется текстовый файл, содержащий несколько строк символов. Подсчитать число символов «-» в этих строках.
10. Задана строка символов. Подсчитать число слов в этой строке. Считать, что слова разделяются одним из символов « » (пробел), «,» (запятая), «.» (точка).
11. Пусть имеется текстовый файл, содержащий несколько предложений. Подсчитать количество предложений и слов в этом файле.
12. Пусть имеется текстовый файл, содержащий несколько слов. Отсортировать эти слова в алфавитном порядке и записать их в другой текстовый файл.
13. Написать программу замены данных в строке. Пусть:
A = "123456789";

B = "67";

C = "-Шестьдесят семь-";

Необходимо найти символы **"67"** (из строки B) и заменить их на **Шестьдесят семь-** (из строки C) в строке A, где A в итоге должна содержать **"12345-Шестьдесят семь-89"**.

Таблица 4.1

Функции работы со строками

Методы класса String	Описание метода
s.length()	Возвращает длину строки S
s.substr(pos, length)	возвращает подстроку из строки S, начиная с номера pos длиной length символов;
s.empty()	возвращает значение true , если строка s пуста, false — в противном случае;
s.insert(pos, s1)	вставляет строку s1 в строку S, начиная с позиции pos ;
s.remove(pos, length)	удаляет из строки S подстроку length длиной pos символов;
s.find(s1, pos)	возвращает номер первого вхождения строки s1 в строку S, поиск начинается с номера pos , параметр pos может отсутствовать, в этом случае поиск идет с начала строки;
s.findfirst(s1, pos)	возвращает номер первого вхождения любого символа из строки s1 в строку S, поиск начинается с номера pos , который может отсутствовать.

§ 5 Динамические типы данных

1. Напишите программу, реализующую объявление, заполнение и удаление динамического массива. Программа также должна выполнять вывод массива на экран и запись его в текстовый (бинарный) файл.
2. Реализуйте предыдущую задачу с помощью подпрограмм (процедур и функций).
3. Дана динамическая матрица случайных чисел размерности $N \times N$ ($N > 9$). Вычислите произведение всех элементов матрицы, у которых индексы строк и столбцов четные. Результат выведите на экран.
4. Описать структуру с именем **STUDENT**, содержащую следующие поля:
 - **NAME** – фамилия и инициалы;
 - **GROUP** – номер группы;
 - **SES** - успеваемость (массив из пяти элементов).

Реализовать программу, используя указатели на структуру. Запишите данные для 10 студентов в файл.

5. Создать структуру «Товар». Каждый товар должен иметь не менее 8 полей, например, название; описание; страна и город, где произведен товар; предприятие-производитель; категория товара (продукты, хозтовары, промтовары и т.д.); цена; вес и т.д. Заполнить динамический массив десятью товарами. Реализовать поиск в массиве по названию, по вхождению слов в описание и по диапазону цены товара.
6. Объявите указатель на массив типа **double** и предложите пользователю выбрать его размер. Далее напишите четыре функции: первая должна выделить память для массива, вторая – заполнить ячейки данными, третья – показать данные на экран, четвертая – освободить занимаемую память. Для обхода массива использовать указатели (запрещено обращаться к элементам массива по индексам).

§ 6 Рекурсия

1. Напишите программу определения факториала "!" для неотрицательных целых чисел, используя рекурсивную функцию.
2. Возведение числа n в степень k — это умножение числа n на себя k раз. Напишите рекурсивную функцию с именем **power()**, которая в качестве аргументов принимает значение типа **double** для n и значение типа **int** для k и возвращает значение типа **double**. Напишите функцию **main()**, которая запрашивает у пользователя ввод аргументов для функции **power()**, и отобразите на экране результаты ее работы.
3. Написать функцию, вычисляющую биномиальный коэффициент C_n^k без использования операторов цикла.
4. Напишите рекурсивную функцию для вычисления суммы первых n элементов целочисленного динамического массива.
5. Написать функцию, вычисляющую **НОД**(a, b) для неотрицательных целых a и b (без циклов).
6. Написать функцию, печатающую цифры десятичного представления своего неотрицательного целого параметра, разделяя их пробелами: а) в обычном порядке; б) в обратном порядке (то и другое — без циклов).
7. Написать функцию, проверяющую правильность скобочной структуры, без циклов: допускаются только символы «(», «)» и «.» (последний означает конец строки); допускаются три вида скобок («()», «[]» и «{}»). Конец строки — «.»

§ 7 Функции и указатели

§ 7.1 Указатели

1. С одномерным массивом, состоящим из n вещественных элементов, выполнить следующее: Преобразовать массив таким образом, чтобы сначала располагались все положительные элементы, а потом – все отрицательные (элементы, равные 0, считать положительными).
2. С одномерным массивом, состоящим из n вещественных элементов, выполнить следующее: Преобразовать массив таким образом, чтобы сначала располагались все элементы, целая часть которых лежит в интервале $[a, b]$, а потом – все остальные.
3. С одномерным массивом, состоящим из n вещественных элементов, выполнить следующее: Преобразовать массив таким образом, чтобы сначала располагались все отрицательные элементы, а потом – все положительные (элементы, равные 0, считать положительными).
4. С одномерным массивом, состоящим из n вещественных элементов, выполнить следующее: Преобразовать массив таким образом, чтобы сначала располагались все элементы, целая часть которых не превышает 1, а потом – все остальные.
5. С одномерным массивом, состоящим из n вещественных элементов, выполнить следующее: Преобразовать массив таким образом, чтобы сначала располагались все элементы, отличающиеся от максимального не более чем на 20%, а потом – все остальные.
6. С одномерным массивом, состоящим из n вещественных элементов, выполнить следующее: Заменить все отрицательные элементы массива их модулями и изменить порядок следования элементов в массиве на обратный.
7. С одномерным массивом, состоящим из n вещественных элементов, выполнить следующее: Сжать массив, удалив из него одинаковые элементы. Освободившиеся в конце массива элементы заполнить нулями.
8. С одномерным массивом, состоящим из n вещественных элементов, выполнить следующее: Сжать массив, удалив из него все элементы, модуль которых не превышает 1. Освободившиеся в конце массива элементы заполнить нулями.

9. С одномерным массивом, состоящим из n вещественных элементов, выполнить следующее: Сжать массив, удалив из него все элементы, модуль которых находится в интервале $[a, b]$. Освободившиеся в конце массива элементы заполнить нулями.
10. С одномерным массивом, состоящим из n вещественных элементов, выполнить следующее: Преобразовать массив таким образом, чтобы сначала располагались все элементы, равные нулю, а потом – все остальные.
11. С одномерным массивом, состоящим из n вещественных элементов, выполнить следующее: Преобразовать массив таким образом, чтобы в первой его половине располагались элементы, стоявшие в нечетных позициях, а во второй половине – элементы, стоявшие в четных позициях.
12. С одномерным массивом, состоящим из n вещественных элементов, выполнить следующее: Преобразовать массив таким образом, чтобы сначала располагались все элементы, модуль которых не превышает 1, а потом – все остальные.
13. С одномерным массивом, состоящим из n вещественных элементов, выполнить следующее: Преобразовать массив таким образом, чтобы элементы, равные нулю, располагались после всех остальных.
14. С одномерным массивом, состоящим из n вещественных элементов, выполнить следующее: Преобразовать массив таким образом, чтобы в первой его половине располагались элементы, стоявшие в четных позициях, а во второй половине – элементы, стоявшие в нечетных позициях.
15. С одномерным массивом, состоящим из n вещественных элементов, выполнить следующее: Сжать массив, удалив из него все элементы, величина которых находится в интервале $[a, b]$. Освободившиеся в конце массива элементы заполнить нулями.

§ 7.2 Подпрограммы-функции

1. Напишите функцию, которая возвращает большее значение из введенных пользователем.
2. Напишите программу, содержащую функцию, которая возводит число a в степень b . Причем a и b вводятся с клавиатуры.
3. Напишите функцию, вычисляющую процент от числа. Например: *321% от числа 3 равен 9.63*.
4. Сделайте программу, функция которой сравнивает введенные числа и результат выдает в виде знаков ">", "<" или "=".

5. Написать функцию, вычисляющую корни квадратного уравнения. В качестве аргументов она принимает коэффициенты (a, b, c) , а возвращает значение по обстоятельству (x_1 и x_2 , либо «Корней нет», либо $= 0$ «Введены некорректные данные»).
6. Напишите функцию, которая возвращает 1, если пользователь ввел гласную букву латинского алфавита, и 0 в противном случае.
7. Написать функцию, специализированную на вывод строки из звездочек, количество которых определяется пользователем.
8. Написать и протестировать функцию, которая из заданного массива формирует новый массив, состоящий только из элементов, дважды входящих в первый массив.
9. Написать и протестировать функцию, возвращающую номер самого последнего элемента из массива, который совпадает с заданным с клавиатуры числом. Если такого элемента нет, функция должна возвращать -1 .

§ 7.3 Функции и указатели

1. Создайте программу, реализующую работу с динамическим массивом. Разработайте 4 функции: первая - инициализация массива с выделением памяти под массив, вторая – заполнение массива данными, третья – вывод данных на экран, четвертая – освобождение занимаемой массивом памяти.
2. В целочисленном динамическом массиве **$x[20]$** определить сумму положительных элементов, делящихся на 5 без остатка, и поставить ее на место максимального элемента массива **$y[10]$** . Реализуйте в виде отдельных функций: 1) создание массивов; 2) поиск элементов массива **$x[20]$** ; 3) замена соответствующего элемента массива **$y[10]$** ; 4) освобождение занимаемой массивами памяти.
3. Программа. Описать функцию $f(x, n, p)$, определяющую, чередуются ли положительные и отрицательные элементы в целочисленном динамическом массиве **$x[n]$** из n элементов и вычисляющую целочисленное значение p . Если элементы чередуются, то p – это сумма положительных элементов, иначе p – это произведение отрицательных элементов. С помощью этой функции провести анализ целочисленного массива **$x[50]$** .
4. Создайте динамический массив случайных чисел. Перемешать его элементы случайным образом так, чтобы каждый элемент оказался на новом месте. Реализуйте программу в виде отдельных подпрограмм-функций.
5. Дан динамический массив символов. Показать номера символов, совпадающих с последним символом строки. Реализуйте программу в виде отдельных подпрограмм-функций.

6. Создайте программу, реализующую работу с динамическим массивом. Разработайте 5 функций: первая - инициализация массива с выделением памяти под массив; вторая – заполнение массива случайными числами; третья – вывод данных на экран; четвертая – сортировка массива таким образом, чтобы первая половина массива была отсортирована по возрастанию, а вторая по убыванию; пятая – освобождение занимаемой массивом памяти.
7. Напишите функцию поиска в динамическом массиве количества чисел, соседи у которых отличаются более чем в 2 раза.
8. Напишите подпрограмму-функцию, определяющую, образует ли элементы динамического массива в данном порядке арифметическую или геометрическую прогрессии.
9. Напишите функцию поиска в динамическом массиве максимального количества одинаковых элементов. Выведите на экран значение этого элемента, их количество и номера в массиве.
10. Напишите функцию, осуществляющую циклический сдвиг динамического массива на k единиц вправо, если первый наименьший элемент массива расположен раньше последнего наибольшего элемента массива, и влево, если иначе.
11. В данном динамическом массиве каждый элемент равен 0, 1 или 2. Переставить элементы массива так, чтобы сначала располагались все нули, затем все единицы и, наконец, все двойки. Дополнительный массив не использовать. Реализуйте алгоритм в виде отдельной подпрограммы-функции

§ 7.4 Функции и файлы

1. Напишите функцию поиска в массиве количества чисел, соседи у которых отличаются более чем в 2 раза. Реализуйте в программе считывание данных (массива) из файла.
2. Напишите подпрограмму-функцию, определяющую, образует ли элементы массива в данном порядке арифметическую или геометрическую прогрессии.
3. Напишите функцию поиска в массиве максимального количества одинаковых элементов. Выведите на экран значение этого элемента, их количество и номера в массиве.
4. Написать функцию, специализированную на вывод строки из звездочек, количество которых определяется пользователем. Написать функцию, вычисляющую биномиальный коэффициент C_n^k . Результаты вычислений запишите в файл.
5. Написать и протестировать функцию, которая из заданного массива формирует новый массив, состоящий только из элементов, дважды входящих в первый массив. Реализуйте в программе считывание и запись данных (массива) в файл.

6. Напишите программу, которая считывала бы саму себя и выводила бы на экран дисплея исходный текст программы в обратном порядке.
7. Имеется файл с текстом. Осуществить шифрование данного текста в новый файл. Осуществить расшифровку полученного текста.

§ 8 Объектно-ориентированное программирование

§ 8.1 Классы

1. Создать класс **Employee**. Класс должен включать, помимо имени и фамилии, поле типа **int** для хранения номера сотрудника и поле типа **float** для хранения величины его оклада. Методы класса должны позволять пользователю вводить и отображать данные класса. Написать функцию **main()**, которая запросит пользователя ввести данные для трех сотрудников и выведет полученную информацию на экран.
2. Создать класс типа круг. Поля-данные: радиус, координаты центра. Функции-члены вычисляют площадь, длину окружности, устанавливают поля и возвращают значения. Функции-члены установки полей класса должны проверять корректность задаваемых параметров (не равны нулю и не отрицательные).
3. Создать класс типа время с полями: час (0–23), минуты (0–59), секунды (0–59). Класс имеет конструктор. Функции-члены установки времени, получения часа, минуты и секунды, а также две функции-члены печати: печать по шаблону «16 часов 18 минут 3 секунды» и «4 p.m. 18 минут 3 секунды». Функции-члены установки полей класса должны проверять корректность задаваемых параметров.
4. Создать класс типа дата с полями: день (1–31), месяц (1–12), год (целое число). Класс имеет конструктор. Функции-члены установки дня, месяца и года, функции-члены получения дня, месяца и года, а также две функции-члены печати: печать по шаблону «5 января 1997 года» и «05.01.1997». Функции-члены установки полей класса должны проверять корректность задаваемых параметров.
5. Создать класс одномерный массив целых чисел (вектор) с полями — количество фактических элементов, массив (динамический). Функции-члены: обращения к отдельному элементу массива, вывода массива на экран, поэлементного сложения и вычитания со скаляром, вывода элемента по заданному индексу.
6. Создать класс множество **Set**. Функции-члены реализуют добавление и удаление элемента, пересечение и разность множеств.

7. Создайте класс **Number**. Добавьте внутри класса функцию (метод) ввода переменной с клавиатуры и функцию вывода данной переменной на экран. Организуйте конструктор и деструктор с соответствующим выводом на экран сообщений «Сработал конструктор!» и «Сработал деструктор!».
8. Измените предыдущую программу таким образом, чтобы класс **Number** состоял из двух полей: целочисленной и символьной переменных. Организуйте работу деструктора и 4 конструкторов: конструктор без параметров, конструктор с целочисленным параметром, конструктор с символьным параметром, конструктор с обоими параметрами.
9. Создайте класс **Children**, который содержит такие поля (члены класса): закрытые (**private**) – имя, отчество и фамилию ребенка, а также его возраст; публичные (**public**) – методы ввода данных и отображения их на экран. Объявить два объекта класса, внести данные и показать их. Организуйте конструктор и деструктор с соответствующим выводом на экран сообщений «Сработал конструктор!» и «Сработал деструктор!».
10. Создать класс типа параллелепипед. Поля – высота, длина и ширина. Функции-члены вычисляют площадь и объем, сумму длин всех ребер параллелепипеда и длину главной диагонали, устанавливают поля и возвращают значения. Функции-члены установки полей класса должны проверять корректность задаваемых параметров (не равны нулю и не отрицательные). Организуйте два вида конструктора: без параметров и с параметрами по умолчанию, а также деструктор с сообщением об уничтожении объекта.
11. Создайте класс «Книга», содержащий следующие поля: название, количество страниц, год издания, цена. Методы: вычисления средней стоимости страницы; сколько лет книге; определение количества дней, прошедших после года издания книги. Создайте для данного класса конструктор и деструктор.
12. Создайте класс одномерный динамический массив **Array**, который содержит такие поля (члены класса): публичные – методы ввода данных и отображения их на экран, а также определение максимального элемента массива. Создайте для данного класса конструктор и деструктор.
13. Создайте класс **Matrix**, который содержит такие поля (члены класса): публичные – методы ввода данных и отображения их на экран, а также определение максимального элемента матрицы. Создайте для данного класса конструктор и деструктор.
14. Определить базовый класс **Автомобиль** с полями *Торговая марка*, *Число цилиндров*, *Мощность*. Создать конструкторы и деструктор объектов, а также метод **Show()**, выводящий информацию об объекте. Определить производный класс Грузовик, добавив в него *характеристику грузоподъемности кузова*. Создать конструкторы объектов производного класса. Переопределить метод **Show()** в производном классе. Создать методы, позволяющие изменять поля объектов базового и производного классов.

15. Создайте класс с именем **CPerson**, содержащий три поля типа **string** для хранения имени, фамилии и отчества. В классе создайте функцию **ShowData()**, выводящую на экран имя, фамилию и отчество. Далее от класса **CPerson** с помощью наследования создайте два класса: **CStudent**, **CProfessor**. К классу **CStudent** добавьте дополнительное поле, содержащее средний бал студента. К классу **CProfessor** три поля: 1) число публикаций профессора, 2) должность (тип - перечисление) - преподаватель, старший преподаватель, доцент, профессор, 3) возраст. Для каждого производного класса переопределите метод **ShowData()**. В основной программе определите массив (можно не динамический) указателей на объекты класса **CPerson**. Далее в цикле нужно организовать ввод студентов и профессоров вперемешку. Когда ввод будет закончен, нужно вывести информацию с помощью метода **ShowData()** обо всех людях.
16. Создайте два базовых класса **ClassX** и **ClassY**, содержащих такие поля (члены класса): защищенные (**protected**) – переменная вещественного типа (**x** для класса **ClassX** и **y** для класса **ClassY**); публичные (**public**) – методы ввода данных и отображения их на экран. Создайте производный класс **ClassZ**, содержащий публичный (**public**) метод – метод расчета произведения **x*y**.
17. В предыдущей задаче организуйте работу конструктора и деструктора.
18. Создайте базовый класс **Human** и класс-наследник **Student**. Класс **Human** описывает модель человека. В нем хранятся имя и фамилия, дата рождения, адрес прописки. Конструктор **Student** принимает все аргументы конструктора базового класса, а также дополнительные аргументы для расширения функционала, такие, как список оценок студента по предметам. Класс **Student** содержит методы вычисления среднего балла студента и вывода его на экран.
19. Разработать три класса, которые следует связать между собой, используя наследование: (1) класс **Product**, который имеет три элемент-данных — имя, цена и вес товара (базовый класс для всех классов); (2) класс **Buy**, содержащий данные о количестве покупаемого товара в штуках, о цене за весь купленный товар и о весе товара (производный класс для класса **Product** и базовый класс для класса **Check**); (3) класс **Check**, не содержащий никаких элемент-данных. Данный класс должен выводить на экран информацию о товаре и о покупке (производный класс для класса **Buy**); Для взаимодействия с данными классов разработать **set**- и **get**-методы. Все элемент-данные классов объявлять как **private**.
20. Класс **Покупатель**: Фамилия, Имя, Отчество, Адрес, Номер кредитной карточки, Номер банковского счета; Конструктор; Методы: установка значений атрибутов, получение значений атрибутов, вывод информации. Создать массив объектов данного класса. Вывести список покупателей в алфавитном порядке и список покупателей, у которых номер кредитной карточки находится в заданном диапазоне.

21. Создайте класс с именем **Train**, содержащий поля: название пункта назначения, номер поезда, время отправления. Ввести данные в массив из пяти элементов типа **Train**, упорядочить элементы по номерам поездов. Добавить возможность вывода информации о поезде, номер которого введен пользователем. Добавить возможность сортировки массива по пункту назначения, причем поезда с одинаковыми пунктами назначения должны быть упорядочены по времени отправления.

§ 8.2 Шаблоны функций

1. Написать программу для нахождения максимального значения в массиве из целочисленных, вещественных, символьных и строковых величин с использованием шаблона функции.
2. Написать шаблонную функцию, которая примет два числа, определит максимальное из них и вернет его в программу. Будем иметь в виду, что в функцию мы можем передать числа разных типов. Возможен и случай, что одно число будет целым, а второе – вещественным.
3. Написать программу для определения суммы значений в массиве (целочисленных/вещественных) с использованием шаблона функции.
4. Реализовать шаблонную функцию, которая вычисляет процент от числа и возвращает значение в программу. И число, и процент передаются как параметры.
5. С использованием шаблона функции написать программу сортировки методом ”пузырька” для массивов величин с целочисленным, вещественным, символьным и строковыми типами данных. Добавить шаблон функции для вывода массивов на экран.
6. Написать программу, которая проверяла бы условие равенства/неравенства двух введенных чисел и выводила бы соответствующее сообщение на экран. В программе необходимо объявить и определить шаблон функции **neq()**, которая будет проверять на неравенство значения различных типов, включая комплексные числа.

§ 8.3 Векторы

1. Задан массив из целых чисел. Переместить все минимальные элементы в начало массива, не меняя порядок других.
2. Задан целочисленный массив. Определить процентное содержание элементов, превышающих среднеарифметическое всех элементов массива.
3. Выполнить сортировку массива по возрастанию (убыванию).

4. Дан массив из 10 элементов. Первые 4 упорядочить по возрастанию, последние 4 по убыванию.
5. Дан массив 20 целых чисел на отрезке $[-2; 5]$. Упорядочить массив, удалив нули со сдвигом влево.
6. Дан массив 15 целых чисел на отрезке $[-5; 5]$. Упорядочить массив, удалив повторяющиеся элементы.
7. Ввести два массива действительных чисел. Определить максимальные элементы в каждом массиве и поменять их местами.
8. Дана квадратная матрица. Вывести на экран элементы, стоящие на диагонали.
9. Дана матрица. Вывести на экран все нечетные столбцы, у которых первый элемент больше последнего.
10. Дана матрица $N \times M$ случайных чисел. Отсортировать элементы главной диагонали матрицы по убыванию.
11. Дана матрица $N \times M$ случайных чисел. Упорядочить первый столбец матрицы по возрастанию, а последний столбец – по убыванию.

§ 8.4 Алгоритмы STL

1. Написать программу, которая с помощью алгоритмов STL **[sort()]** выполняет сортировку массива строк.
2. Отсортируйте по возрастанию (убыванию) и выведите на экран массив вещественных чисел. Для решения этой задачи воспользуйтесь заголовочным файлом **[#include <functional>]** и объектами-функциями **[greater/less]**, напр., **[sort(array, array+array_size, greater<double>())]**.
3. Написать программу, которая с помощью алгоритмов STL **[next_permutation()]** выводит на экран все перестановки строки “abcd”.
4. Написать программу поиска подстроки в строке, основываясь на функциях стандартных библиотек **[find()]**.
5. В заданной строке (**StrText**) заменить все вхождения заданной подстроки (**StrFind**) на заданную строку (**StrReplace**). Например, **StrText="Informational Technology"**, **StrFind="Technology"**, **StrReplace="System"**.
6. Основываясь на функциях стандартных библиотек, написать программу подсчета количества каждого слова, встречающегося в тексте.

7. Написать шаблонную функцию **input**, вводящую вектор с клавиатуры.
8. Написать шаблонную функцию **part**, принимающую вектор и возвращающую а) первую половину, б) среднюю треть его элементов (тоже как вектор). Написать шаблонную функцию **concat**, принимающую два вектора и возвращающую их сцепление (сначала идут элементы первого, а затем второго).
9. Написать шаблонную функцию **repeat**, принимающую вектор **v** и неотрицательное целое число **n** и возвращающую новый вектор, полученный повторением вектора **v** **n** раз.
10. Написать шаблонную функцию **subseq**, принимающую два вектора и проверяющую, что второй из них является подпоследовательностью первого (например, для векторов (1, 2, 3, 4, 5, 6, 7, 8, 9) и (1, 5, 7, 8) ответ будет **true**, а для (1, 2, 3, 4, 5, 6, 7, 8, 9) и (2, 1, 5, 7, 8) ответ будет **false**).
11. Написать шаблонную функцию **enlarge**, принимающую вектор по ссылке и вставляющую между каждыми соседними элементами их полусумму.
12. Написать функцию **shorten**, принимающую вектор из целых чисел по ссылке и удаляющую элементы, стоящие между 1 и 2 (именно в таком порядке).
13. Написать шаблонную функцию **co**, принимающую два вектора и возвращающую число вхождений второго вектора, как подпоследовательность из элементов, идущих подряд (а не как в задаче 3), в первый (с использованием алгоритма **search**).
14. Написать набор шаблонных функций для реализации понятия множества, которое хранится как упорядоченный вектор. Должны быть реализованы следующие операции: ввод, вывод, добавить элемент, удалить элемент, проверить принадлежность элемента множеству, объединение, пересечение и разность.
15. Написать набор шаблонных функций для реализации понятия многочлена одной переменной, который хранится как вектор коэффициентов. Должны быть реализованы следующие операции: ввод, вывод, задать постоянный многочлен, задать многочлен x , вычислить значение многочлена, сумму, произведение, неполное частное и остаток от деления многочленов (последние две функции предполагают, что коэффициенты принадлежат полю, т. е. можно делить на любой ненулевой коэффициент, и притом точно).

§ 9 Общие задачи

1. Вычисление числа π . Для вычисления числа π используем ряд

$$\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \dots$$

Провести вычисления, обеспечив заранее заданную точность $\epsilon > 0$. При этом вычисления заканчиваются при $a < \epsilon$, где a – разность между текущим приближенным значением числа π и значением **const double PI** (значение числа π с 15 правильными знаками после запятой).

2. Напишите программу, которая находит факториал от введенного числа. Реализуйте алгоритм в виде рекурсивной функции.
3. Определить число e — основание натуральных логарифмов с помощью ряда:

$$e = 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \frac{1}{4!} + \dots + \frac{1}{n!}.$$

Вычислить значения данного ряда для всех значений n от 1 до 20. Для каждого случая вывести на экран n и соответствующее значение .

4. Дан текстовый файл f , компоненты которого являются целыми числами. Записать в файл g все четные числа файла f , а в файл h – все нечетные. Порядок следования чисел сохраняется.
5. Ввести натуральное N и проверить, является ли оно совершенным? Примечание: совершенное число равно сумме всех своих делителей, исключая само число. Например, $6 = 1 + 2 + 3$.
6. Составить программу для нахождения всех автоморфных чисел в отрезке $[m, n]$. Автоморфным называется целое число, которое равно последним числам своего квадрата. Например: $52 = 25$, $62 = 36$, $252 = 625$.
7. Известно, что сумма N первых нечетных чисел равна квадрату числа N . Например, $1 + 3 + 5 = 3^2$, $1 + 3 + 5 + 7 = 4^2$ и т.д. Ввести натуральное K и распечатать таблицу всех натуральных чисел от 1 до K и их квадратов с использованием указанного соотношения.

8. Дано натуральное число n ($n \leq 100$), определяющее возраст человека (в годах). Дать для этого числа наименования «год», «года», или «лет». Например, 1 год, 23 года, 45 лет и т.д.
9. Написать программу вычисления методом Монте-Карло площади фигуры, ограниченной половиной синусоиды.
10. Задача на перебор. Получить все перестановки элементов 1, ..., 6.
11. Написать программу для вычисления методом Монте-Карло площади S тела, ограниченного кривыми $xy = a$ и $x + y = \frac{5}{2}a$. Сравнить результат с точным значением.
12. Игра «Угадай число». Один из играющих задумывает число от 1 до 1000, другой пытается угадать его за десять вопросов вида: верно ли, что задуманное число больше такого-то числа. Написать программу, играющую за отгадчика.
13. Пусть даны четыре целых числа ($hour, min, sec, time$). Первые три из них ($hour, min, sec$) – это время запуска ракеты в часах, минутах и секундах. Четвертое ($time$) — определяет время полета в секундах. Вычислить время возвращения ракеты на землю.
14. Один из простейших способов шифровки текста состоит в табличной замене каждого символа другим символом — его шифром. Выбрать некоторую таблицу, разработать способ ее представления, затем
 - а) зашифровать данный текст;
 - б) расшифровать данный текст.
15. Численно решить уравнение радиоактивного распада:

$$\frac{dN}{dt} = -\lambda N.$$

Разработать алгоритм решения задачи и написать программу на языке программирования C++. Сравнить численное решение с аналитическим. Определить условия сходимости.

16. Создать типизированный файл записей со сведениями о телефонах абонентов; каждая запись имеет поля: фамилия абонента, год установки телефона, номер телефона. По заданной фамилии абонента выдать номера его телефонов. Определить количество установленных телефонов с N -го года. Отсортировать список по алфавиту и вывести все записи на экран.
17. Напишите функцию, которая преобразовывает значение, заданное в радианах, в значение, выраженное в градусах, угловых минутах и угловых секундах. Воспользуйтесь указателем на структурный тип данных.

18. Напишите программу, которая считывает числовые значения из файла, вычисляет значение полусуммы наибольшего и наименьшего элементов, а затем подсчитывает количество значений, не превышающих по величине полусумму, и больших чем полусумма.
19. Напишите программу, которая меняет местами столбцы матрицы, содержащие наибольший и наименьший элементы.
20. Треугольник задан координатами трех своих вершин. Определить, где находится точка с указанными координатами - внутри или вне треугольника.
21. Составьте алгоритм и напишите программу для вычисления приближенного значения натурального логарифма от произвольного значения аргумента $|x| < 1$, вводимого с клавиатуры. Ряд Тейлора для этой функции имеет вид:

$$\ln(1 + x) = x - \frac{x^2}{2!} + \frac{x^3}{3} - x^4 + \dots$$

22. Напишите программу, которая "сжимает" текстовый файл, считывая его элементы и заменяя все повторяющиеся символы, например, **ссссс**.... текстом **с(n)**, где **n** – число повторений символа **с**. В программе используйте процедуры-функции.

Ответы и указания

§ 1.1. Одномерные массивы

1.

```
#include <iostream>

using namespace std;

int main()
{
    int n=10;
    double x[n];
    double sum=0.0;
    for (int i=0; i<n; i++)
    {
        cout << "Please, input x[" << i << "]="; cin >> x[i];
        if (x[i]<0) sum+=x[i];
    }
    cout << "Summa = " << sum;
    return 0;
}
```

2.

```
#include <iostream>
#include <ctime>
#include <cstdlib>

using namespace std;

int main()
{
    double x[10];
    srand(time(0));
    for (int i=0; i<n; i++)
    {
        x[i]=1.0+rand()%100;
        cout << x[i] << "\n";
    }
    cout << endl;
    for (int i=0; i<n; i++)
    {
```



```

        if (i%2!=0) cout << i << "\n";
    }
    return 0;
}

```

§ 1.2. Сортировка и упорядочение массивов

1.

```

#include <iostream>
#include <cstdlib>
#include <ctime>
using namespace std;

int main()
{
    srand(time(0));
    const int n=5, m=6;
    int x[n][m];
    for (int i=0; i<n; i++)
    {
        for (int j=0; j<m; j++)
        {
            x[i][j]=1+rand()%10;
            cout << x[i][j]<<"\t";
        }
        cout << "\n";
    }
    return 0;
}

```

§ 4. Обработка строк

1.

```

#include<iostream>
#include<cstring>
using namespace std;
int main()
{
    char str[50]="(5+3)*7+65+7896";
    cout << "Initial string is: " << str << endl;
    for (int i=0; i<strlen(str); i++)
        if (str[i] == '+')
            str[i] = '-';
    cout << "Final string is: " << str << endl;
    return 0;
}

```

3.

```
#include <iostream>
#include <string>
using namespace std;
int main()
{
    setlocale(LC_ALL, "Russian");
    string s;
    cout << " Введите строку: \n"; cin >> s;
    cout << " Строка " << s << " содержит "
        << s.length() << " символ(а').\n";
    return 0;
}
```

4.

```
#include <iostream>
#include <string>
using namespace std;
int main()
{
    setlocale(LC_ALL, "Russian");
    string str("Выведите на экран номер символа в строке.");
    string symbol="э";
    unsigned int pos=str.find(symbol);
    if ((pos>=0) && pos(<str.length()))
        cout << "Номер первого вхождения символа ("
            << symbol << "') в строке \n\n("
            << str << "')\n\n равен: "<< pos << endl;
    else cout << "Такого символа в строке нет!\n";
    return 0;
}
```

§ 5. Динамические типы данных

1.

```
#include <iostream>
#include <fstream>
#include <cstdlib>

using namespace std;
int main()
{
    ofstream out("Array.txt");
    const int n=10; double* x;
    x = new double[n];
    for (int i=0; i<n; i++)
    {
```

```

        *(x+i)=rand()%10;
        cout << "x["<< i << "]= " << *(x+i) << "\t";
        out << *(x+i) << "\n";
    }
    cout << "\n"; delete[] x; out.close();

    ifstream in("Array.txt");
    double * y;
    y = new double[n];
    for (int i=0; i<n; i++)
    {
        in >> *(y+i);
        cout << "y["<< i << "]= " << *(y+i) << "\t";
    }
    delete[] y; in.close(); return 0;
}

```

6. dd

```

#include <iostream>
#include <fstream>
#include <cstdlib>
using namespace std;

double* init(int n);
void data(int n, double* x);
void print(int n, double* x);
void write_file(int n, double* x);
void del(double* x);

int main()
{
    int n; cout << "Input n: "; cin >> n;
    double* x=init(n);
    data(n,x);
    print(n,x);
    write_file(n,x);
    del(x);
    return 0;
}

double* init(int n) { return new double[n]; }
void data(int n, double* x) { for (int i=0; i<n; i++) x[i]=rand()%
10; }
void print(int n, double* x) { for (int i=0; i<n; i++) cout << x[i
] << "\t"; }
void write_file(int n, double* x)
{
    ofstream out ("Array.txt");
    for (int i=0; i<n; i++) out <<"x[" << i <<"]=" << x[i] << "\n"
;
    out.close();
}

```

```

}
void del(double *x) { delete[] x; }

```

§ 8.1. Классы

7.

```

#include <iostream>
using namespace std;
class Number
{
    int a;
public:
    Number()
    { cout << "Сработал конструктор без параметров" << "\n"; }
    Number(int A)
    {
        a = A;
        cout << "Сработал конструктор с параметром: " << "\n";
        cout << "a= " << a << "\n";
    }
    void set_Number()
    { cout << "Введите целое число a= "; cin >> a; }
    void out_Number()
    { cout << "Число a= " << a << "\n"; }
    ~Number()
    { cout << "\nСработал деструктор" << "\n"; }
};
int main()
{
    setlocale(0, "rus");
    cout << "\n***Первый объект***" << "\n";
    Number obj1;
    obj1.set_Number();
    obj1.out_Number();

    cout << "\n***Второй объект***" << "\n";
    Number obj2(100);
    obj2.out_Number();
    return 0;
}

```

16.

```

#include <iostream>
using namespace std;
class ClassX
{
    protected:

```

```

        double x;
    public:
        void setX()
        {cout << "Input x: "; cin >> x;}
        void outX()
        {cout << "x= " << x << "\n";}
};
class ClassY
{
    protected:
        double y;
    public:
        void SetY()
        {cout << "Input y: "; cin >> y;}
        void outY()
        {cout << "y= " << y << "\n";}
};
class ClassZ: public Class X, public Class Y
{
    public:
        int make_xy() { return x*y; }
};
int main()
{
    classZ obj;
    obj.setX(); obj.setY();
    obj.outX(); obj.outY();
    cout << "xy= " << obj.make_xy() << "\n";
    return 0;
}

```

§ 9. Общие задачи

2.

```

#include <iostream>
#include <stdlib.h>

using namespace std;
int factorial (int num);

int main()
{
    int number=5;
    cout << number << "!=" << factorial(number);
    return 0;
}
int factorial (int num)
{

```

```
    if (num<0)
    {
        cout << "Error!"; exit(1);
    }
    else if (num==0)
        return 1;
    return num*factorial(num-1);
}
```

Темы проектных работ

1. Программно реализовать игру "Крестики-нолики". Играют либо два игрока, либо один игрок с компьютером.
2. Программно реализовать игру "Морской бой". Играют либо два игрока, либо один игрок с компьютером.
3. Программно реализовать игру "Шашки". Играют либо два игрока, либо один игрок с компьютером.
4. Программно реализовать игру "Шахматы". Играют либо два игрока, либо один игрок с компьютером.
5. Программно реализовать игру "Игра в города" с компьютером. Указание: Необходимо создать файл с базой данных городов.
6. Программно реализовать игру "Жизнь".
7. Программно реализовать игру "Игра в 15".
8. Программно реализовать игру "Тетрис".
9. Программно реализовать игру "Угадай число". Играют либо два игрока, либо один игрок с компьютером.
10. Программно реализовать игру "Утраченный клад".

Список рекомендуемой литературы

1. Стивен, П. *Язык программирования C++. Лекции и упражнения* / П. Стивен. - Киев: Диалектика-Вильямс, 2018. - 1244 с.
2. Шилдт, Г. *C++. Полное руководство* / Г. Шилдт. - Киев: Вильямс, 2019. - 800 с.
3. Конова, Е. А. *Алгоритмы и программы. Язык C++: учебное пособие* / Е. А. Конова, Г. А. Поллак. - Санкт-Петербург: Лань, 2020. - 384 с.
4. Окулов, С. М. *Основы программирования: учебное пособие* / С. М. Окулов. - М.: Лаборатория знаний, 2020. - 339 с.
5. Андриянова, А. А. *Алгоритмизация и программирование. Практикум: учебное пособие* / А. А. Андриянова, Л. Н. Исмагилов, Т. М. Мухтарова. - Санкт-Петербург: Лань, 2019. - 240 с.
6. Златопольский, Д. М. *Программирование: типовые задачи, алгоритмы, методы* / Д. М. Златопольский. - М.: Лаборатория знаний, 2020. - 226 с.