

Лабораторная работа 1.

Строки.

Часть 1.

Выполните следующие задачи по ссылке:

<https://informatics.msk.ru/mod/statements/view.php?id=88217#1>

Часть 2.

Регулярное слово – слово, состоящее только из больших латинских букв.

Палиндром – это слово, которое одинаково читается слева направо и справа налево.

Алфавитный порядок задается таблицей ASCII.

Задание 1.

Выделить в строке-предложении *s* все слова, разделенные символами-разделителями «`_.,;:\n\t!?`». Обработать выделенные слова в соответствии с вариантом.

1. Напечатать все слова, начинающиеся на большую и заканчивающиеся на маленькую букву.
2. Напечатать все слова, содержащие хотя бы одну цифру.
3. Напечатать все слова, содержащие хотя бы одну маленькую латинскую букву.
4. Напечатать все слова, содержащие хотя бы одну большую латинскую букву.
5. Напечатать все слова, состоящие только из маленьких латинских букв.
6. Напечатать все слова, состоящие только из больших латинских букв.
7. Напечатать все слова, состоящие только из цифр.
8. Напечатать все слова, состоящие не менее чем из четырех букв.
9. Напечатать все пятибуквенные слова.
10. Напечатать все слова, содержащие буквы и цифры.
11. Напечатать все слова, не содержащие ни одной цифры.
12. Напечатать все слова, не содержащие ни одной большой латинской буквы.
13. Напечатать все слова, содержащие хотя бы две цифры.
14. Напечатать все слова, содержащие две рядом стоящие буквы.
15. Напечатать все слова, содержащие две рядом стоящие цифры.
16. Напечатать все слова, состоящие не более чем из пяти букв.
17. Напечатать все слова, состоящие только из латинских букв.

Задание 2.

Выделить в строке-предложении *s* все слова, разделенные символами-разделителями «`_.,;:\n\t!?`». Обработать выделенные слова в соответствии с вариантом.

В1. Подсчитать количество слов, начинающихся на большую букву и содержащих хотя бы один арифметический знак. Напечатать все слова, содержащие две рядом стоящие одинаковые буквы.

В2. Подсчитать количество регулярных слов. Напечатать в перевернутом виде все слова, которые содержат два экземпляра заданного символа.

В3. Напечатать все слова, начинающиеся с большой буквы. Напечатать самое длинное регулярное слово, которое состоит из одинаковых символов.

В4. Напечатать слово, содержащее наибольшее количество цифр. Напечатать количество слов, содержащих хотя бы два арифметических знака.

В5. Напечатать все регулярные слова. Напечатать в перевернутом виде самое длинное слово, состоящее только из цифр и латинских букв.

В6. Найти количество слов, содержащих более одной цифры, и, исключив все арифметические знаки из этих слов, напечатать их. Напечатать в порядке возрастания все числа, встретившиеся в словах.

В7. Определить количество слов, которые содержат заданное подслово и хотя бы одну цифру, и напечатать их. Напечатать в порядке убывания все числа, встретившиеся в словах.

В8. Определить количество слов, содержащих и буквы, и цифры, и арифметические знаки. Напечатать их. Напечатать все симметричные слова, содержащие наибольшее количество цифр.

В9. Подсчитать количество регулярных слов, содержащих хотя бы две одинаковые буквы. Напечатать все слова, имеющие одну цифру, удалив из таких слов все арифметические знаки.

В10. Найти самое длинное регулярное слово и удалить из него все гласные буквы. Найти все слова, в которых имеются либо только цифры, либо только латинские буквы.

В11. Подсчитать количество слов, состоящих из одинаковых букв или одинаковых цифр. Напечатать в перевернутом виде слова, имеющие хотя бы один арифметический знак.

В12. Напечатать все слова, которые начинаются с большой буквы и заканчиваются заданным двухбуквенным подсловом. Определить количество

слов, содержащих согласные латинские буквы, и напечатать порядковые номера этих слов.

B13. Напечатать все слова, имеющие в своем составе согласные латинские буквы. Определить количество слов, которые не имеют в своем составе ни одной цифры, и напечатать эти слова.

B14. Напечатать все симметричные слова, предварительно удалив из них цифры. Напечатать все слова, состоящие только из согласных латинских букв.

B15. Найти все слова, содержащие числа в диапазоне от 10 до 100, и подсчитать их сумму. Напечатать слова, не имеющие цифр, предварительно удалив арифметические знаки.

B16. Подсчитать количество слов, начинающихся с большой буквы и оканчивающихся цифрой. Напечатать слова, содержащие заданное под-слово и хотя бы один арифметический знак.

B17. Подсчитать количество слов, содержащих хотя бы одну согласную латинскую букву и хотя бы одну цифру. Напечатать все слова, состоящие только из четных цифр, и подсчитать сумму этих цифр.

B18. Напечатать все слова, которые содержат хотя бы один арифметический знак и заканчиваются на цифру. Определить количество слов, содержащих все маленькие латинские гласные буквы.

B19. Найти количество симметричных регулярных слов и напечатать их. Напечатать в перевернутом виде все слова, содержащие согласные латинские буквы.

B20. Найти и напечатать все слова, содержащие наибольшее количество букв, если только буквы расположены в алфавитном порядке. Подсчитать количество симметричных слов, имеющих более двух арифметических знаков.

Задание 3.

С помощью текстового редактора создать файл `text1.txt` и заполнить словами из латинских букв (не менее 30 слов).

Файл `number1.txt` программно заполнить случайными целыми числами, принадлежащими диапазону от -100 до 100. Количество чисел в файле задается с клавиатуры. Количество чисел в каждой строке является произвольным числом из диапазона от 0 до 10 и задается с помощью функции `rand()`. Вывести числа на консоль, сохранив разбиение на строки, соответствующее файлу `number1.txt`.

При реализации программы необходимо для заполнения файла `number1.txt` случайными целыми числами использовать заголовочный файл,

содержащий функцию для заполнения файла. При работе с файлом, заполненным словами, можно использовать функции работы со строками. При выводе результата в файл number3.txt сохранить разбиение на строки.

Записать в файл total.txt сначала все элементы файла number2.txt, выровненные по правому краю, а затем все элементы файла text2.txt, выровненные по центру. Переименовать файл total.txt (новое имя файла вводится с клавиатуры).

Вывести на консоль сначала все элементы файла number2.txt, выровненные по правому краю, а затем все элементы файла text2.txt, выровненные по центру.

Варианты

1. Найти сумму максимальных элементов файла number1.txt. Все числа, меньшие первого максимального, вывести в файл number3.txt и на консоль. В конец файла number2.txt добавить слова "Result = ", номер строки и максимальное число этой строки, вывести полученный результат на консоль. Вывести все слова файла text1.txt, начинающиеся с гласных букв латинского алфавита. Результат записать в файл text2.txt.
2. Подсчитать, число элементов файла number1.txt, кратных 9. Вывести все числа, кратные 9, в файл number3.txt и на консоль. В конец файла number2.txt добавить слова "Result = " и число элементов, вывести полученный результат на консоль. Вывести все строки файла text1.txt, начинающиеся с заглавной буквы. Результат записать в файл text2.txt.
3. Подсчитать, сколько раз каждое число встречается в тексте файла number1.txt. Все встречающиеся числа и их количество вывести в файл number3.txt и на консоль. В конец файла number2.txt добавить слова "Result = " и количество различных чисел в тексте, вывести полученный результат на консоль. В словах файла text1.txt заменить заданную букву на букву, вводимую с клавиатуры. Результат записать в файл text2.txt.
4. Найти максимальное число в каждой строке файла number1.txt. Вывести в файл number3.txt и на консоль все числа файла number1.txt, стоящие до найденного максимального. В конец файла number2.txt добавить слова "Result = " и найденное максимальное число, вывести полученный результат на консоль. Вывести все слова файла text1.txt, начинающиеся с согласных букв латинского алфавита. Результат записать в файл text2.txt.

5. Найти сумму чисел каждой строки файла number1.txt. Вывести в файл number3.txt и на консоль номер строки и полученную сумму. В конец файла number2.txt добавить слова "Result = " и количество строк файла number1.txt, вывести полученный результат на консоль. Вывести все слова файла text1.txt, начинающиеся и оканчивающиеся одной и той же буквой. Результат записать в файл text2.txt.
6. Преобразовать в логарифм все неотрицательные элементы файла number1.txt. Результат вывести в файл number3.txt и на консоль. В конец файла number2.txt добавить слова "Result = " и количество четных неотрицательных элементов, вывести полученный результат на консоль. Все слова файла text1.txt, заканчивающиеся на букву, заданную с клавиатуры, вывести в файл text2.txt.
7. Подсчитать факториал каждого положительного элемента файла number1.txt. Полученные элементы вывести в файл number3.txt и на консоль. В конец файла number2.txt добавить слова "Result = " и число положительных элементов, вывести полученный результат на консоль. Удалить все пустые строки файла text1.txt. Результат записать в файл text2.txt.
8. Найти разность пар соседних элементов файла number1.txt. Результат вывести в файл number3.txt и на консоль. В конец файла number2.txt добавить слова "Result = " и количество элементов, равных нулю, вывести полученный результат на консоль.

Вывести все слова файла text1.txt, предварительно преобразовав каждое из них по следующему правилу: оставить в слове только первые вхождения каждой буквы. Результат записать в файл text2.txt.

9. Возвести в квадрат все отрицательные элементы файла number1.txt. Результат вывести в файл number3.txt и на консоль. В конец файла number2.txt добавить слова "Result = " и число нечетных отрицательных элементов, вывести полученный результат на консоль. Вывести все слова файла text1.txt, которые содержат определенное количество заданной буквы. Результат записать в файл text2.txt.
10. Каждый элемент файла number1.txt заменить на экспоненту в степени, равной этому элементу. Результат вывести в файл number3.txt и на консоль. В конец файла number2.txt добавить слова "Result = " и число элементов, больших единицы, вывести полученный результат на консоль. Каждое слово файла text1.txt циклически сдвинуть вправо на половину своей длины. Например, сдвиг слова asdhjk приводит к hjkasd. Результат записать в файл text2.txt.
11. Уменьшить каждый элемент строки файла number1.txt на число, соответствующее номеру этой строки. Результат вывести в файл number3.txt и на консоль. В конец файла number2.txt добавить слова "Result = " и количество отрицательных элементов, вывести полученный результат на консоль. Вывести различные слова файла text1.txt. Результат записать в файл text2.txt.
12. Извлечь квадратный корень из каждого положительного элемента файла number1.txt. Результат вывести в файл number3.txt и на консоль. В конец файла number2.txt добавить слова "Result = " и число положительных элементов, вывести полученный результат на консоль. Удалить из файла text1.txt все подслова, содержащие последовательность трех букв, вводимых с клавиатуры. Результат записать в файл text2.txt.
13. Найти логарифм каждого элемента файла number1.txt, большего единицы. Все элементы, меньшие единицы, вывести в файл number3.txt и на консоль. В конец файла number2.txt добавить слова "Result = " и число элементов, больших единицы. Самые длинные и самые короткие слова файла text1.txt вывести в файл text2.txt.

14. Подсчитать произведение отрицательных элементов файла number1.txt. Все неотрицательные элементы вывести в файл number3.txt и на консоль. В конец файла number2.txt добавить слова "Result = " и найденное произведение, вывести полученный результат на консоль.
Вывести все строки файла text1.txt, начинающиеся со строчной буквы. Результат записать в файл text2.txt.
15. Найти разность элементов соседних строк файла number1.txt. Все полученные элементы вывести в файл number3.txt и на консоль. В конец файла number2.txt добавить слова "Result = " и число элементов кратных 5, вывести полученный результат на консоль.
Все слова файла text1.txt заменить на обратные (например, kot заменить на tok), вывести в файл text2.txt.
16. Подсчитать произведение четных элементов файла number1.txt. Все нечетные элементы вывести в файл number3.txt и на консоль. В конец файла number2.txt добавить слова "Result = " и найденное произведение, вывести полученный результат на консоль.
Все слова файла text1.txt, начинающиеся с заданной буквы (вводится с клавиатуры), вывести в файл text2.txt.
17. Умножить каждый элемент файла number1.txt на номер соответствующей ему строки. Результат вывести в файл number3.txt и на консоль. В конец файла number2.txt добавить слова "Result = " и количество элементов, кратных 11, вывести полученный результат на консоль.
Каждую строку файла text1.txt циклически сдвинуть вправо на n символов. Например, сдвиг строки asdfghjk на 3 символа: hjkasdfg. Результат записать в файл text2.txt.
18. Найти сумму пар соседних элементов файла number1.txt. Результат вывести в файл number3.txt и на консоль. В конец файла number2.txt добавить слова "Result = " и количество четных элементов, вывести полученный результат на консоль.
Зашифровать слова в текстовом файле text1.txt по принципу: каждую гласную продублировать, а каждую согласную заменить на следующую согласную в слове. Например, слово "строка" после шифрования будет выглядеть как "тркоосаа". Результат записать в файл text2.txt.
19. Подсчитать произведение элементов каждой строки файла number1.txt. Все числа, кратные 10, вывести в файл number3.txt и на консоль. В конец файла number2.txt добавить слова "Result = ", номер строки и найденные произведения. Вывести полученный результат на консоль.
После каждой гласной слова файла text1.txt добавить последовательность символов, заданную с клавиатуры. Результат записать в файл text2.txt.
20. Подсчитать, сколько раз каждое положительное число встречается в каждой строке файла number1.txt. Номер строки, все встречающиеся числа и их количество вывести в файл number3.txt и на консоль. В конец файла number2.txt добавить слова "Result = " и количество строк в тексте, вывести полученный результат на консоль.
Обратить все строки файла text1.txt и вывести их в файл text2.txt.

Задание 4.

Сгенерируйте и обработайте JSON-файл событий. **Сторонними библиотеками пользоваться запрещено**, реализовать чтение/запись файла стандартными средствами работы с C++.

JSON (JavaScript Object Notation) – это легковесный формат обмена данными, который легко читается и пишется людьми и легко анализируется и генерируется машинами.

1. Формат JSON:

- JSON представляет данные в форме пар ключ-значение. Каждая пара разделена запятой. Это напоминает словари в Python или ассоциативные массивы в других языках программирования.
- JSON поддерживает следующие типы данных: числа, строки, булевы значения (true/false), массивы, объекты (сложные структуры данных) и null.

2. Структура JSON:

- **Объекты JSON:** Заключаются в фигурные скобки {} и представляют собой набор пар ключ-значение. Ключи должны быть строками, а значения могут быть любого из поддерживаемых типов данных.
- **Массивы JSON:** Заключаются в квадратные скобки [] и представляют собой упорядоченный список значений (которые могут быть разных типов).

3. Работа с JSON в C++:

- Основные операции включают парсинг JSON из строки или файла, доступ к данным через ключи, модификацию данных и сериализацию объектов обратно в строку JSON или файл.

4. Прикладное использование:

- JSON часто используется для хранения конфигураций, обмена данными между сервером и клиентом в веб-приложениях, хранения сложных структурированных данных (например, информации о пользователях, продуктах, транзакциях) и для многих других задач, где требуется легко читаемый и гибкий формат обмена данными.

5. Примеры JSON:

- Простой объект: {"name": "John", "age": 30}
- Массив: ["apple", "banana", "cherry"]
- Вложенные структуры: {"name": "John", "address": {"street": "Main", "city": "New York"}}

Задачи:

1. Генерация Отчета о Продажах:

- Написать программу, генерирующую JSON файл с данными о продажах товаров за месяц (название товара, количество, цена).

2. Анализ Данных Погодной Станции:

- Программа для чтения JSON файла с данными о температуре, влажности и давлении от погодной станции и анализа средних значений.

3. Управление Базой Данных Сотрудников:

- Создать программу для добавления, удаления и изменения информации о сотрудниках компании в JSON файле (имя, должность, зарплата).

4. Трекер Задач:

- Разработать систему управления задачами, где задачи и их статусы хранятся в JSON формате.

5. Библиотечный Каталог:

- Программа для генерации и обновления каталога книг библиотеки в JSON формате (автор, название, год издания).

6. Расписание Транспорта:

- Создать JSON файл с расписанием общественного транспорта (номер маршрута, время отправления, интервалы).

7. Анализ Результатов Опроса:

- Программа для анализа данных опроса, хранящихся в JSON файле (вопросы, выборы ответов, статистика ответов).

8. Учет Финансов:

- Разработать систему для ведения учета личных финансов, где транзакции хранятся в JSON файле.

9. Инвентаризация Оборудования:

- Генерация и обновление файла JSON с информацией об офисном оборудовании (наименование, количество, состояние).

10. Управление Заказами в Ресторане:

- Создание JSON файла для учета заказов в ресторане (блюдо, количество, стоимость).

11. Система Бронирования Отелей:

- Разработать программу для чтения и обновления информации о бронированиях в отеле в формате JSON.

12.Трекер Здоровья:

- Программа для хранения и анализа данных о здоровье (вес, калории, шаги) в формате JSON.

13.Каталог Рецептов:

- Создание JSON файла с каталогом кулинарных рецептов (ингредиенты, инструкции, время приготовления).

14.Учет Запасов на Складе:

- Программа для учета товарных запасов на складе с использованием JSON (наименование товара, количество, дата поступления).

15.Система Отслеживания Посылок:

- Разработать систему для отслеживания статуса посылок, используя JSON файлы (номер отслеживания, статус, предполагаемая дата доставки).

16.Регистрация Участников Мероприятия:

- Создание и обновление списка участников мероприятия в формате JSON (имя, контактная информация, статус регистрации).

17.Журнал Посещений Фитнес-Центра:

- Программа для ведения журнала посещений фитнес-центра с данными в JSON (имя клиента, дата и время посещения).

18.Каталог Автомобилей:

- Генерация и обновление JSON файла с информацией о автомобилях в автосалоне (марка, модель, год выпуска).

19.Анализ Трафика Веб-Сайта:

- Разработать программу для анализа логов веб-сайта, хранящихся в JSON формате (IP-адреса, страницы, время посещения).

20.Управление Расписанием Занятий:

- Создать систему для управления расписанием учебных занятий в JSON формате (предмет, преподаватель, время).

