// C++ · Лабораторная 04

Строки

Обработка символов, поиск, преобразование слов и проверка границ строки.

Цель и порядок работы

Освоить основные алгоритмы работы со строками и методы тестирования программ.

В работе нужно обрабатывать введённый текст: находить символы и группы, считать или преобразовывать их. До написания программы опишите ход решения словами: что считается словом, какие символы подходят под условие и что делать, если искомого фрагмента нет.

1. Выполните задания своего варианта и добавьте оглавление.
2. До программирования составьте словесное описание решения, затем блок-схему.
3. Реализуйте алгоритм на C++, скомпилируйте и отладьте программу.
4. Проверьте нормальные, граничные и некорректные данные, покрытие решений и логических условий.
5. Сохраните исходную строку, результат и тесты в отчёте.

Требования к коду, заготовка отчёта, тестирование и блок-схемы →

Скачать текущие задания и варианты+ в PDF

Исходные материалы: Методичка №5 · PDF

Перед началом: Лекция 4 · строки · Лекция 4 · char и ASCII · Лекция 3 · создание функций · Лекция 2 · циклы

Задания по вариантам

Выполните все задания назначенного варианта.

Общие требования к вариантам

Используйте std::string и контейнеры из лекции 4, функции из лекции 3 и циклы из лекции 2. C-строки, char* и ручное выделение памяти не требуются. Строки до 200 ASCII-символов; пробелы сохраняйте при чтении через std::getline.

Если не сказано иначе, слово — максимальная непустая последовательность непробельных символов. В заданиях на арифметическое выражение числа от 1 до 1000; промежуточные результаты по модулю не больше 10^9, деление вещественное, делитель ненулевой. Скобок и унарных знаков в выражении нет.

Где запрещены рабочие строки, изменяйте исходную строку операциями erase, insert и перестановкой символов; отдельный результат допускается только там, где это разрешено условием. Для ASCII-гласных считайте a, e, i, o, u без учёта регистра.

Для поиска возрастающей последовательности сравнивайте коды ASCII, при равной длине выбирайте первую. Для групп одинаковых символов считайте максимальные серии. Для прогрессий выводите максимальные непрерывные серии не короче трёх чисел.

Вариант 1

Задание 1

Найти в строке все сдвоенные символы. Вывести их на экран и подсчитать их количество.

Считайте максимальные серии одинаковых символов длиной ровно 2; серия из трёх и более не является сдвоенным символом.

Задание 2

Из заданной строки удалить все последовательности символов, находящиеся между корректно записанными круглыми скобками (скобки удалить тоже). Скобки могут быть вложенными.

Для удаления скобочных фрагментов сначала проверьте баланс скобок. Некорректную строку не преобразовывайте; выведите понятное сообщение.

Вариант 2

Задание 1

Определить, корректно ли расставлены в строке круглые и квадратные скобки. Скобки могут быть вложенными.

Задание 2

Переставить в начало строки все буквы в порядке их следования, затем цифры, и в конце – остальные символы. Рабочие строки не использовать.

Проверяйте не только число скобок, но и порядок типов: ([)] некорректно. Используйте стек открывающих скобок; закрывающая должна соответствовать последней открытой. Преобразование второго задания независимо от проверки первого.

Вариант 3

Задание 1

Строка – это правильное арифметическое выражение, содержащее только натуральные числа и знаки арифметических операций между ними (+, -, *, /). Подсчитать результат, выполняя арифметические операции в порядке их следования.

Задание 2

В строке, содержащей не менее двух слов, разделенных пробелом, поменять местами первое и последнее слова.

Вариант 4

Задание 1

Строка – это правильное арифметическое выражение, содержащее только натуральные числа и знаки арифметических операций между ними (+, -, *, /). Подсчитать результат, выполняя арифметические операции в порядке их приоритета.

Задание 2

В строке, содержащей не менее трех слов, разделенных пробелом, поменять местами второе и предпоследнее слова.

Вариант 5

Задание 1

В строке, содержащей несколько слов, разделенных пробелом, переставить слова в обратной последовательности.

Задание 2

В строке определить самую длинную последовательность символов, стоящих по возрастанию.

Вариант 6

Задание 1

В строке, содержащей несколько слов, разделенных пробелом, перевернуть каждое слово.

Задание 2

Строка представляет собой последовательность натуральных чисел, разделенных пробелом. Найти в ней все последовательности (не менее трех чисел), являющиеся арифметическими прогрессиями.

Вариант 7

Задание 1

Ввести две строки, состоящие только из нулей и единиц. Считая их изображениями двоичных чисел, сложить их и вывести сумму на экран в двоичной и десятичной системах счисления. Исходные слагаемые в десятичную систему счисления не преобразовывать.

Задание 2

Строка представляет собой последовательность слов, разделенных пробелом. Удалить из нее все повторения слов. Рабочие строки не использовать.

Каждое двоичное слагаемое содержит от 1 до 30 бит. Двоичную сумму получайте сложением битов с переносом; только готовую сумму преобразуйте в long long. Ведущие нули допустимы; нулевой результат выводится как 0.

Вариант 8

Задание 1

В заданной строке заменить все последовательности одинаковых символов длиной более двух на конструкцию (k)a, где k- количество повторений символа a.

Задание 2

В заданной строке найти первую и последнюю цифры. Отсортировать символы, стоящие между ними, по возрастанию. Рабочие строки не использовать.

Вариант 9

Задание 1

В заданной строке найти минимальное и максимальное целое число (числа могут быть отрицательными, в этом случае знак “минус” перед ними ставится без пробела). Предусмотреть случай, когда в строке чисел нет.

Задание 2

В заданной строке найти подстроку, находящуюся между пробелами с заданными номерами. Рабочие строки не использовать. Предусмотреть случай, когда в строке таких пробелов нет.

Вариант 10

Задание 1

В заданной строке, состоящей из нескольких слов, разделенных пробелами, оставить по одному пробелу между словами. Предусмотреть случай, когда в строке пробелов нет.

a*bc -&19 -l sb3> 4-5 z%x

Задание 2

Вывести все 4-х символьные сочетания, состоящие из символов заданной строки, удовлетворяющие заданным правилам их формирования. Правила задать с терминала для каждого i-го символа (i=1,...,4): 1-й символ – это буква или цифра? Если буква, то гласная или согласная? Если цифра, то четная или нечетная? 2-й символ – это буква или цифра? И так далее. Ответы: 1 – да, 0 – нет. Использованный символ строки далее в новые сочетания не входит.

Сочетания формируйте жадно: для каждой позиции выбирайте первый ещё не использованный подходящий символ исходной строки. После полного сочетания его символы считаются использованными. Если новое полное сочетание не собрать, остановитесь.

Вариант 11

Задание 1

Строка состоит из нескольких слов, разделенных пробелом. Определить минимальное количество первых символов слов, по которым можно различить слова в строке.

Задание 2

В заданной строке переставить слова по возрастанию их длины.

Если есть одинаковые слова, сообщите, что различить их по префиксу невозможно. При равных длинах порядок слов сохраняйте.

Вариант 12

Задание 1

Строка состоит из натуральных чисел, разделенных пробелом, и диапазонов чисел (начальное число-тире-конечное число). Удалить из строки все числа, входящие в эти диапазоны. Оставшиеся числа вывести на экран по возрастанию (диапазоны не выводить).

Задание 2

Удалить из строки все повторения символов. Рабочие строки не использовать.

Вариант 13

Задание 1

Задана строка, в которой имеются цифры. Вывести на экран названия встретившихся цифр словами в порядке их следования.

Задание 2

Строка состоит из натуральных чисел, разделенных пробелом. Определить цифру, встречающуюся в этих числах наибольшее количество раз.

Названия цифр выводите по-английски: zero, one, two, three, four, five, six, seven, eight, nine. Пример a2b0 → two zero. Если цифр нет, сообщите об этом.

Вариант 14

Задание 1

Задана строка, в которой имеются наименования цифр. Вывести на экран эти цифры в порядке их следования.

Задание 2

Строка состоит из натуральных чисел, разделенных пробелом. Определить цифры, не входящие ни в одно их этих чисел.

Наименования — отдельные английские слова zero, one, two, three, four, five, six, seven, eight, nine в нижнем регистре. Остальные слова пропускайте. Пример one cat zero → 1 0.

Вариант 15

Задание 1

Задана строка, представляющая собой уравнение с одним неизвестным в следующем формате: ax+b=c где a, b и c – целые числа, положительные или отрицательные. Определить значение неизвестного x.

Для нулевого коэффициента при x различайте отсутствие решений и бесконечно много решений. Параметры по модулю до 1000.

Задание 2

Строка состоит только из букв. Переставить их так, чтобы гласные и согласные чередовались. Оставшиеся лишними гласные или согласные переместить в конец строки. Рабочие строки не использовать.

$$ax + b = c, \quad x = \frac{c - b}{a} \quad (a \neq 0)$$

Вариант 16+ · Сжатие серий и восстановление

Кодирование

Для ASCII-строки из латинских букв замените каждую серию длиной больше 2 записью (k)a. Для остальных серий сохраните буквы. Пустая строка допустима.

Обратное преобразование+

Напишите функцию восстановления такой записи. Отклоняйте неполные скобки, количество меньше 3 и отсутствие буквы после скобки. Для строки до 100 символов проверьте точное совпадение после двух преобразований.

Вариант 17+ · Очистка вложенных скобок

Разбор

Проверьте правильность круглых и квадратных скобок через stack. Остальные ASCII-символы не влияют на баланс. При ошибке выведите её позицию; незакрытые скобки после конца обозначьте отдельно.

Преобразование+

Только для корректной строки удалите все фрагменты в круглых скобках вместе со скобками, включая вложенные. Квадратные фрагменты снаружи круглых сохраняйте.

Вариант 18+ · Числа в тексте и частоты

Разбор

Из ASCII-строки выделите все целые числа: необязательный минус непосредственно перед цифрами и максимальная последовательность цифр. Числа по модулю не больше 1000000; другие символы — разделители.

Дополнение+

Через map выведите частоты, минимум, максимум и сумму чисел. Отсутствие чисел — отдельный исход. Объясните разбор текста a-12b003--4: числа −12, 3, −4.

Вариант 19+ · Слова и устойчивый порядок

Нормализация

Считайте строку до 200 ASCII-символов. Словом считайте непустую последовательность непробельных символов. Удалите начальные и конечные пробелы и оставьте между словами ровно один.

Дополнение+

Удалите повторные слова, сохранив первое вхождение. Затем упорядочьте оставшиеся слова по длине; при равных длинах сохраните прежний порядок. Напишите собственный простой алгоритм перестановки через циклы.

Вариант 20+ · Двоичное сложение и проверка

Сложение

Введите две непустые двоичные строки длиной до 30. Проверьте каждый символ, выполните сложение справа налево с переносом, не преобразуя слагаемые в числа заранее.

Дополнение+

Выведите двоичный результат без лишних ведущих нулей, число переносов и десятичное значение результата в long long. Проверьте нули, разные длины и перенос, создающий новый разряд.

Возможные вопросы к защите

Попробуйте ответить своими словами и показать соответствующее место в своей программе. Под вопросом можно открыть ориентир для ответа и перейти к нужной теме лекции.

▼ Как связаны char, код ASCII и символ цифры?

Объясните отличие '5' от 5 и получение значения цифры через разность с '0'.

Повторить: Лекция 4 · таблица ASCII

▼ Как обойти строку через for и изменить её символы?

Покажите обход по индексу и присваивание новому значению символа. Ссылки для этого не обязательны.

Повторить: Лекция 4 · строки · Лекция 2 · циклы

▼ Как работают find, substr, insert и erase у std::string?

Приведите пример из своего решения, объясните позицию, длину фрагмента и std::string::npos.

Повторить: Лекция 4 · строки

▼ Какие индексы допустимы и что проверяет at()?

Отдельно рассмотрите пустую строку и обращение к последнему обычному символу.

Повторить: Лекция 4 · строки

▼ Почему operator>> не подходит для чтения целой строки с пробелами?

Сравните форматированное чтение и std::getline; вспомните оставшийся перевод строки.

Повторить: Лекция 1 · ввод и вывод · Лекция 4 · строки

▼ Какие случаи проверяют правильность поиска и удаления подстроки?

Предложите тесты: не найдено, начало, конец, несколько и пересекающиеся совпадения.

Повторить: Лекция 4 · строки · Лекция 5 · диагностика
