

Обработка последовательности литер входного текстового файла.
Простейшие приёмы лексического анализа. Диаграммы состояний и переходов.

Составить программу на языке Си, выполняющую анализ и обработку вводимого текста в соответствии с выданным преподавателем вариантом задания. В программе не должно быть ограничений на количество и длину строк исходного текста. Конец файла должен обрабатываться корректно, как это делается утилитами UNIX (*cat*, *wc* –с и т.д.).

При разработке алгоритма рекомендуется использовать диаграммы Тьюринга или другие подходящие модели, и помещать соответствующие иллюстрации в отчет (пункт «идея, метод, алгоритм решения»).

Систематическое решение задач лексического анализа обычно осуществляется с помощью конечных автоматов, семантика состояний–переходов которых соответствует концепции машин Тьюринга. Для обозначения состояний удобен перечислимый тип.

Кроме того, в системе UNIX имеются инструментальные средства построения лексических и синтаксических анализаторов (*lex/yacc*, *gnu flex/bison*, *pccts*, ...), которые изучаются в курсе «Системное и прикладное программное обеспечение/Системы программирования» на 2 курсе.

Рекомендуется предварительно отработать ввод/вывод программы на прототипе – программе дублирования текстового файла и сравнить ее с командой Unix *ис –с* для тестовых файлов.

Полезными для этой работы являются библиотечные функции стандартной библиотеки Си: *isalpha*, *isdigit*, *isspace*, *isalnum*, *isprint*, *tolower*, *toupper*. Тем, кто уже знаком с описанием и использованием функций в языке Си, рекомендуется реализовать их самостоятельно.

Терминология: **слово** – произвольная (в данном задании *непустая*) последовательность литер, не являющихся разделителями; **число** – произвольная непустая последовательность цифр данной системы счисления, быть может, начинающаяся со знака + или – и представляющая собой отдельное слово; **разделители слов**: пробелы, запятая, табуляции, начало и конец строки и файла; **строка** – последовательность литер, заканчивающаяся символом перевода строки '\n' в языке Си, за исключением последней строки, которая может завершаться признаком конца файла (константой *EOF*, возвращаемой функцией чтения!). При чтении текстового файла переход на новую строку, если она существует (!), в языке Си осуществляется как очередная операция чтения, в то время, как на Паскале для этого требуется вызвать специальную процедуру перевода строки *readln*.; **текст** – последовательность строк.

Замечания

1. Обратите внимание на различие кодов окончания строк в ОС UNIX, Windows и MacOS (см. материал ЛР №3)!
2. Не следует использовать кодозависимые константы для обработки конца строки/файла.
3. Необходимо соблюдать стандарты Си ANSI X3.159 или ISO 9899, задавая ключи *-std=<стандарт, c99 или c89>* и – *pedantic* или, для стандарта 1989 г., ключ *-ansi*, вместе с ключом *-Wall* (для выдачи предупреждающих сообщений).
4. В языке Си состояние конца файла (значение, возвращаемое функцией *feof(FILE *)*, устанавливается *только после* обращения к файлу на чтение или запись.
5. В отличие от Паскаля (и C++!) тип литерной константы в языке Си – *int*, благодаря чему русские буквы в однобайтовых кодировках представлены положительными числами.

Варианты заданий

1. Подсчитать число слов в однострочных комментариях (//) в программе на Си.
2. Подсчитать число слов в многострочных комментариях (* и */) в программе на Си.
3. Подсчитать число слов в комментариях ({ и } или (* и *)) в программе на Паскале.
4. Подсчитать количество *полнотетрадных* (со всеми необходимыми ведущими нулями) двоичнокодированных десятичных чисел (*BCD – Binary Coded Decimal*).
5. Подсчитать количество слов с лексикографически возрастающими буквами латинского алфавита.
6. Подсчитать количество слов – правильных идентификаторов языков Си или Паскаль.
7. Закодировать текст по Цезарю с переменным ключом, равным номеру буквы в слове + 3.
8. Раскодировать текст, закодированный по Цезарю с переменным ключом, равным номеру слова в строке.
9. Напечатать значения пятиричных чисел, не являющихся троичными числами, в десятичной системе.
10. Отделить пробелом слова, находящиеся вплотную к концу строки и к началу следующей.
11. Выделить все 11-ричные числа с лексикографическим возрастанием цифр.
12. Ко всем допустимым целым числам добавить ведущие нули.
13. У всех допустимых 16-ричных чисел перенести ведущие нули в младшие разряды слова фиксированной длины.
- 14.* Подсчитать количество существительных в тексте на немецком языке.
15. Подсчитать контрольные суммы всех слов исходного текста.
16. Подсчитать количество малобуквенных слов во всех строках исходного текста.
17. Выделить все числа, записанные в троичной системе счисления и кратные 3.
18. Перевести все мерные температуры из шкалы Цельсия (C) в шкалу Фаренгейта (F). Например: 100C → 212F.
19. Перевести все мерные длины из дюймов (in) в миллиметры (мм). Например: 10in → 254мм.
20. Перевести все мерные расстояния из миль (mi) в километры (км). Например: 1000mi → 1609км.
- 21.*** Подсчитать количество слов, которые являются числами в римской системе счисления.
22. Подсчитать количество слов, содержащих не менее 3-х букв.
23. Подсчитать количество положительных десятичных чисел в строке, допустимых 16-битными процессорами.
24. Подсчитать количество восьмиричных чисел, находящихся в диапазоне от 10 до 1000.
25. Выделить предпоследнее шестнадцатиричное число в строке.
26. Выделить предпоследнее десятичное число и вычислить сумму его цифр.

27. Выделить предпоследнее десятичное число и распечатать его цифры в системах счисления с основаниями 3, 7 и 11.
28. Выделить первые и последние (по счёту) десятичные числа строк текста.
29. Удалить десятичные числа, не превышающие INT_MAX.
- 30.* Удалить десятичные числа, превышающие INT_MAX.
31. Распечатать каждое третье двоичное число.
32. Выделить все десятичные числа от 17 до 77 по модулю и распечатать их значения в словесной форме по-русски.
33. Выделить все восьмеричные числа от 17 до 77 по модулю и распечатать их значения в словесной форме по-английски.
- 34.** Выделить все девятеричные числа от 17 до 77 по модулю и распечатать их значения в словесной форме по-немецки.
- 35.* Выделить все десятичные числа от 17 до 77 по модулю и распечатать их значения в словесной форме по-французски.
- 36.** Выделить все восьмиричные числа от 17 до 77 по модулю и распечатать их значения в словесной форме по-испански.
- 37.** Выделить все девятеричные числа от 17 до 77 по модулю и распечатать их значения в словесной форме по-итальянски.
38. Выделить все шестнадцатиричные числа, имеющие максимальное число цифр в 32-битной архитектуре.
39. Подсчитать количество шестнадцатиричных слов в строке, изображающих отрицательные числа в 64-битной архитектуре.
40. Выделить беззнаковые восьмеричные числа и напечатать их цифры в двоичной системе.
41. Выделить десятичные целые числа без знака и напечатать суммы их четных цифр.
42. Выделить шестнадцатиричные целые числа без знака и напечатать шестнадцатиричную сумму их нечетных цифр.
43. Подсчитать среднюю длину слова в тексте, закодированном в UTF-8.

Варианты заданий составили: Зайцев В.Е., Журавлева Т.Э., Калинин А.Л., Крылов С.С., Лебедев А.В., Сеницкий П.А., Сошников Д.В., Перетягин И.А., Овечкис А.Г.

Лабораторная работа №12 на 2011/12 уч. год: 8 факультет, 1 курс, 1 семестр

Техника работы с целыми числами. Системы счисления.

Составить программу на языке Си в целом типе данных, которая для любых допустимых и корректно записанных чисел этого типа в десятичном изображении, поступающих на стандартный ввод программы, выполняет указанное вариантом действие над их значениями. Нумерация цифр в записи числа ведется справа налево, от младших разрядов к старшим, начиная с нуля. Результатом действия над введенным числом является целое или булевское значение в зависимости от конкретного варианта, направляемое в стандартный вывод. Количество разных тестов для данной задачи должно быть порядка 10. Концом пакета тестов является конец файла. В набор тестов должны быть включены разнообразные, в том числе и простейшие тесты (одноцифренные числа), крайние значения для данной задачи (например, 0), тесты с легко предсказуемым результатом ($0 + 0 = 0$), отрицательные и очень большие числа. При составлении программы и тестов необходимо учесть разрядность целого типа инструментальной (домашней) и целевой (лабораторной) ЭВМ, уточнив её по документации, или предусмотреть её вычисление в начале программы. В процессе составления, отладки и тестирования программы полезно применить технику диаграмм состояний-переходов (см. лаб. работы 6, 9, 11).

Для выполнения задания (помимо конспекта лекций) могут быть полезными книги Д. Кнута «Искусство программирования, т. II», Р. Грэхема и др. «Конкретная математика. Основание информатики», Г. Уоррена, мл. «Алгоритмические трюки для программистов» и П. Нодена и К. Китте «Алгебраическая алгоритмика».

Варианты задания (устанавливаются преподавателем):

0. Проверить палиндромиию числа
1. Удалить среднюю цифру числа
2. Выбрать тройки идущих подряд цифр числа, сумма которых максимальна
3. Выбрать идущие подряд тройки цифр, упорядоченных по возрастанию
4. Выбрать цифры, равные сумме двух предыдущих
5. Выбрать цифры, равные модулю разности двух предыдущих
- 6,7. Вычислить обратный (дополнительный) десятичный код
8. Перенести первый блок нулей из младших разрядов в середину десятичной записи числа
9. Проверить упорядоченность цифр числа по неубыванию
- 10,11. Выполнить циклический сдвиг влево (вправо) десятичного представления числа
- 12,13. Перенести ведущие нули десятичного представления в младшие разряды (в середину двоичной записи числа)
14. Проверить, есть ли одинаковые цифры в смежных разрядах
15. Проверить, все ли цифры в смежных разрядах различны
16. Проверить, есть ли цифры, не встречающиеся ни разу
17. Поменять местами первую и последнюю цифры
18. Поменять местами вторую и предпоследнюю цифры
19. Отсечь первую и последнюю цифры
20. Удалить вторую и предпоследнюю цифры
21. Приписать в начало и конец по единице
22. Удалить все чётные цифры из числа
23. Удалить все цифры на чётных позициях
- 24,25. Инкрементировать/декрементировать (увеличить/уменьшить на единицу) все чётные цифры
26. После каждых двух цифр вставить их сумму, если она также является цифрой
27. После каждых двух цифр вставить абсолютное значение их разности
28. Получить двоично-кодированное десятичное представление числа
29. Получить восьмирично-кодированное десятичное представление
- 30,31. Упорядочить цифры числа попарно по возрастанию (убыванию) I \leftrightarrow II, III \leftrightarrow IV, ...
- 32,33. Проверить, содержит ли число равные количества нулей и единиц в машинном представлении (двоичном виде)