

## Задание 2. Обработка строк символов

**ASCII-строка** — строка, содержащая символы таблицы кодировки ASCII. (<https://ru.wikipedia.org/wiki/ASCII>). Размер строки может быть достаточно большим, чтобы вмещать многостраничные тексты, например, главы из книг, если задача связана с использованием файлов или строк, порождаемых генератором случайных чисел. Тексты при этом могут не нести смыслового содержания. Для обработки в программе предлагается использовать данные, содержащие символы только из первой половины таблицы (коды в диапазоне  $0-127_{10}$ ), что связано с использованием кодировки UTF-8 в ОС Linux в качестве основной. Символы, содержащие коды выше  $127_{10}$ , должны отсутствовать во входных данных кроме оговоренных специально случаев.

**Разработанная программа должна читать обрабатываемый текст из файла и загружать полученные результаты также в файл. Ввод имен входного и выходного файлов должен осуществляться с использованием консоли. Аналогичным образом осуществляет ввод остальных параметров, необходимых для выполнения программы.**

1. Разработать программу, которая «переворачивает» заданную позициями  $N_1-N_2$  часть ASCII-строки символов ( $N_1, N_2$  вводятся как параметры). Преобразование осуществляется внутри введенной строки.
2. Разработать программу, находящую в заданной ASCII-строке первую при обходе **от конца к началу** последовательность  $N$  символов, каждый элемент которой определяется по условию «больше предшествующего» ( $N$  вводится в качестве параметра). Понятие «больше предшествующего» определяется в соответствии с порядком следования символов от начала строки.
3. Разработать программу, находящую в заданной ASCII-строке первую слева направо последовательность  $N$  символов, каждый элемент которой определяется по условию «меньше предшествующего» ( $N$  *вводится как отдельный параметр*). Понятие «меньше предшествующего» определяется в соответствии с порядком следования символов от начала строки.
4. Разработать программу, находящую в заданной ASCII-строке последнюю при перемещении слева направо последовательность  $N$  символов, каждый элемент которой определяется по условию «больше предшествующего» ( $N$  *вводится как отдельный параметр*).

Понятие «больше предшествующего» определяется в соответствии с порядком следования символов от начала строки.

5. Разработать программу, заменяющую все строчные гласные буквы в заданной ASCII-строке заглавными.
6. Разработать программу, заменяющую все согласные буквы в заданной ASCII-строке их **ASCII кодами в десятичной системе счисления**.
7. Разработать программу, заменяющую все гласные буквы в заданной ASCII-строке их **ASCII кодами в шестнадцатиричной системе счисления**. Код каждого символа задавать в формате «0xDD», где **D** — шестнадцатиричная цифра от 0 до F.
8. Разработать программу, заменяющую все цифры (за исключением нуля) в заданной ASCII-строке **римскими цифрами**. То есть, соответствующим комбинациями букв для цифр от 1 до 9.
9. Разработать программу, которая «**переворачивает на месте**» заданную ASCII-строку символов (не копируя строку в другой буфер).
10. Разработать программу, которая меняет на обратный порядок следования символов **каждого слова** в ASCII-строке символов. Порядок слов остается неизменным. Слова состоят только из букв. Разделителями слов являются все прочие символы. Преобразование осуществляется внутри введенной строки.
11. Разработать программу вычисления **отдельно количества гласных и согласных букв** в ASCII-строке. Вывод результатов организовать в файл (используя соответствующие преобразования чисел в строки).
12. Разработать программу, определяющую **минимальный и максимальный (по числовому значению) символ в заданной ASCII-строке**. Вывести в файл символ и его числовой код.
13. Разработать программу, заменяющую все строчные буквы в заданной ASCII-строке прописными, а прописные буквы — строчными.
14. Разработать программу, вычисляющую **отдельно число прописных и строчных букв** в заданной ASCII-строке. Вывод результатов организовать в файл (используя соответствующие преобразования чисел в строки).

15. Разработать программу, которая на основе заданной ASCII-строки символов, решает вопрос, является ли данная строка **палиндромом**.
16. Разработать программу, которая вычисляет **количество цифр и букв** в заданной ASCII-строке. Вывод результатов организовать в файл (используя соответствующие преобразования чисел в строки).
17. Разработать программу, которая **меняет на обратный порядок следование слов** в ASCII-строке символов.
18. Разработать программу, **заменяющую все согласные буквы в заданной ASCII-строке на заглавные**.
19. Разработать программу, вычисляющую число вхождений различных **отображаемых символов** в заданной ASCII-строке. Вывод результатов организовать в файл (используя соответствующие преобразования чисел в строки).
20. Разработать программу, вычисляющую **число вхождений различных цифр** в заданной ASCII-строке. Вывод результатов организовать в файл (используя соответствующие преобразования чисел в строки).
21. Разработать программу, осуществляющую **нахождение суммы всех цифр** в заданной ASCII-строке. Вывод результатов организовать в файл (используя соответствующие преобразования чисел в строки).
22. Разработать программу, вычисляющую **число вхождений различных знаков препинания** в заданной ASCII-строке. Вывод результатов организовать в файл (используя соответствующие преобразования чисел в строки).
23. Разработать программу, которая ищет в ASCII-строке заданную подстроку и возвращает **индекс первого символа первого вхождения подстроки в строке**. Подстрока вводится как параметр. Вывод результатов организовать в файл (используя соответствующие преобразования чисел в строки).
24. Разработать программу, которая ищет в ASCII-строке заданную подстроку и возвращает **список индексов первого символа для**

**всех вхождений** подстроки в строке. Подстрока вводится как параметр. Вывод результатов организовать в файл (используя соответствующие преобразования чисел в строки).

25. Разработать программу, которая определяет в ASCII-строке **частоту встречаемости различных идентификаторов**, являющихся словами, состоящими из букв и цифр, начинающихся с буквы. Разделителями являются все другие символы. Для тестирования можно использовать программы, написанные на различных языках программирования. Вывод результатов организовать в файл (используя соответствующие преобразования чисел в строки).
26. Разработать программу, которая определяет **количество целых чисел** в ASCII-строке. числа состоят из цифр от 0 до 9. Разделителями являются все другие символы. Вывод результатов организовать в файл (используя соответствующие преобразования чисел в строки).
27. Разработать программу, которая определяет **частоту встречаемости (сколько раз встретилось в тексте) пяти ключевых слов языка программирования C**, в произвольной ASCII-строке. Ключевые слова не должны являться частью идентификаторов. Пять искомых ключевых слов выбрать **по своему усмотрению**. Тестировать можно на файлах программ. Вывод результатов организовать в файл (используя соответствующие преобразования чисел в строки).
28. Разработать программу, которая в заданной ASCII-строке определяет **корректность вложенности круглых скобок «(» и «)»**. Необходимо учесть, что вложенные скобки могут образовывать в тексте различные группы. Например: ... (...) ... (...) ....
29. Разработать программу, которая ищет в ASCII-строке **слова — палиндромы и формирует из них новую строку**, в которой слова разделяются пробелами. Слова состоят из букв. Все остальные символы являются разделителями слов.
30. Разработать программу, которая ищет в ASCII-строке слова, **начинающиеся с заглавной буквы и формирует из них новую строку**, в которой слова разделяются пробелами. Слова состоят из букв. Все остальные символы являются разделителями слов.

31. Разработать программу, которая ищет в ASCII-строке **целые числа и формирует из них новую строку**, в которой числа разделяются знаком «+». Числа состоят из цифр от 0 до 9. Все остальные символы являются разделителями чисел (включая и буквы).
32. Разработать программу, которая на основе анализа двух входных ASCII-строк формирует на выходе две другие строки. В первой из выводимых строк содержатся символы, которых нет во второй исходной строке. Во второй выводимой строке содержатся символы, отсутствующие в первой входной строке (**разности символов**). Каждый символ в соответствующей выходной строке должен встречаться только один раз.
33. Разработать программу, которая на основе анализа двух ASCII-строк формирует на выходе строку, содержащую символы, присутствующие в обеих строках (**пересечение символов**). Каждый символ в соответствующей выходной строке должен встречаться только один раз.
34. Разработать программу, которая на основе анализа двух ASCII-строк формирует на выходе строку, содержащую символы, присутствующие в одной или другой (**объединение символов**). Каждый символ в соответствующей выходной строке должен встречаться только один раз.