

Указания к выполнению

Данное задание состоит из четырех частей: посимвольная обработка строк, стандартная библиотека, обработка слов и кодирование*. В каждой части необходимо описать одну или несколько функций.

Задания оформляются одним проектом.

- Все функции следует помещать в одну библиотеку (h-файл + src-файл) и сопровождать комментариями.
- Каждой функции вашей библиотеки должен соответствовать тест в библиотеке тестов.
- Каждый тест должен содержать набор (не менее трёх) тестовых случаев, доказывающих, что функция работает корректно.
- Тесты должны содержать комментарии к каждому тестовому случаю.
- Обработку ошибок необходимо реализовывать с помощью исключений.
- Программа должна сигнализировать о наличии утечек памяти..

Основная программа должна выполнять все тесты и затем переходить к диалогу с пользователем. Диалог должен предлагать выполнить одно из заданий. Каждое задание должно сопровождаться описанием в консоли: что вычисляется и какие параметры может задать пользователь.

Посимвольная обработка строк

При решении задач данного блока следует использовать статические строки `char*` и посимвольную обработку строк.

Использование функций `cstring` не допускается, однако можно использовать функции `cctype`.

1. Посчитать количество символов, отличных от цифр, пробелов и знаков препинания.
2. Посчитать количество предложений в строке. Предложения могут заканчиваться точками, восклицательными или вопросительными знаками.
3. Посчитать наибольшее количество латинских букв, идущих подряд в строке.
4. Посчитать количество слов в строке, которые оканчиваются заданной буквой.
5. Найти последовательность не пробельных символов максимальной длины.
6. Дано целое число K ($0 < K < 256$). Зашифровать текст строки, циклически сдвинув каждый её символ в кодовой таблице на K позиций.
7. Дан символ C . Найти последовательность символов C максимальной длины.
8. Дано положительное целое число N . Вставить между символами строки по N символов «*». Если это не возможно, выдать сообщение об этом.
9. Удалить из строки все пробелы.
10. Проверить, является ли строка правильной записью положительного целого числа в десятичной системе счисления.
11. Дан символ C . Удалить из строки последовательности подряд идущих символов C , оставив один.
12. Дано положительное целое число N . Циклически сдвинуть текст строки на N позиций влево.
13. Заменить все латинские буквы в строке на символ «*».
14. Удалить из строки все знаки препинания.
15. Дан символ C и число K . Проверить, что символ C встречается в строке не более K раз.

16. Проверить, является ли строка правильной записью положительного целого числа в [шестнадцатеричной системе счисления](#).
17. Дано положительное целое число N . Циклически сдвинуть текст строки на N позиций вправо.
18. Дан символ C . Заменить все повторные вхождения символа C в строке на символ «*».
19. Проверить, является ли строка правильной записью положительного целого числа в [восьмеричной системе счисления](#).
20. Вставить в строку символ «*» перед каждой последовательностью пробелов.
21. Найти последовательность максимальной длины из символов, стоящих в лексикографическом порядке.
22. Дан символ C . Посчитать количество слов содержащих заданный символ.
23. Дано положительное целое число N . Удалить из строки все последовательности из N одинаковых символов.
24. Удалить в строке из каждой группы подряд идущих цифр, в которой больше двух цифр, все цифры, начиная с третьей..
25. Проверить, является ли строка правильной записью [идентификатора](#).

Стандартная библиотека

Задания выполнять в двух вариантах:

- 1) с использованием динамических строк `char*` и библиотеки `cstring`;
 - 2) с использованием строк `string`.
-
1. Даны положительные целые числа N_1 и N_2 и строки S_1 и S_2 . Получить из этих строк новую строку, содержащую первые N_1 символов строки S_1 и последние N_2 символов строки S_2 .
 2. Дана строка S , число K и символ C . Удалить из строки все символы, расположенные между $K-1$ и K вхождениями заданного символа.
 3. Даны строки S , S_1 , S_2 . Вставить в строку S перед последним вхождением строки S_1 строку S_2 .
 4. Даны положительные целые числа N_1 и N_2 и строки S_1 и S_2 . Получить из этих строк новую строку, содержащую последние N_1 символов строки S_1 и первые N_2 символов строки S_2 в обратном порядке.
 5. Дана строка S , число K и символ C . Удалить из строки первые K вхождений символа C .
 6. Даны строки S , S_1 , S_2 . Удалить из строки S все символы входящие в строку S_1 и не входящие в строку S_2 . Даны положительные целые числа N_1 и N_2 и строки S_1 и S_2 . Получить из этих строк новую строку, содержащую последние N_1 символов строки S_1 до вхождения в неё строки S_2 и первые N_2 символов после вхождения. Вхождение строки S_2 считать единственным.
 7. Дана строка S , число K и символ C . Поменять порядок символов, расположенных между $K-1$ и K вхождениями заданного символа.
 8. Даны строки S , S_1 , S_2 . Вставить в строку S после первого вхождения строки S_1 строку S_2 .
 9. Даны положительные целые числа N_1 и N_2 и строки S_1 и S_2 . Получить из этих строк новую строку, содержащую последние N_1 символов строки S_1 перед N_2 вхождением строки S_2 .
 10. Дана строка S , число K и символ C . Вставить между каждой парой символов строки по K символов C .
 11. Даны строки S , S_1 , S_2 . Заменить на строку S каждое нечётное вхождение строки S_1 строку S_2 .

12. Дана строка S , число K и символ C . Поменять местами подстроки расположенные до $K-1$ и после K вхождений заданного символа.
13. Даны положительные целые числа N_1 и N_2 и строки S_1 и S_2 . Получить из этих строк новую строку, содержащую первые N_1 символов строки S_1 , не принадлежащих строке S_2 , и последние N_2 символов строки S_2 , не принадлежащих строке S_1 .
14. Дана строка S , число K и символ C . Поменять местами подстроки до и после $K-1$ и K вхождений заданного символа.
15. Даны строки S , S_1 , S_2 . Вставить в строку S перед первым вхождением любого символа строки S_1 строку S_2 .
16. Даны положительные целые числа N_1 и N_2 и строки S_1 и S_2 . Получить из этих строк новую строку, содержащую первые N_1 не пробельных символов строки S_1 и последние N_2 не пробельных символов строки S_2 в обратном порядке.
17. Дана строка S , число K и символ C . Удалить из строки последние K вхождений символа C .
18. Даны строки S , S_1 , S_2 . Заменить в строке S каждое вхождение i -ого символа строки S_1 на i -ый символ строки S_2 .
19. Даны положительные целые числа N_1 и N_2 и строки S_1 и S_2 . Получить из этих строк новую строку, содержащую первые N_1 символов строки S_1 до вхождения в неё строки S_2 и последние N_2 символов после вхождения. Вхождение строки S_2 считать единственным.
20. Дана строка S , число K и символ C . Поменять порядок символов, расположенных до $K-1$ и после K вхождений заданного символа.
21. Даны строки S , S_1 , S_2 . Вставить в строку S после каждого вхождения любого символа строки S_1 строку S_2 .
22. Даны положительные целые числа N_1 и N_2 и строки S_1 и S_2 . Получить из этих строк новую строку, содержащую первые N_1 символов строки S_1 после N_2 и до N_2+1 вхождений строки S_2 .
23. Дана строка S , число K и символ C . Удалить из строки последние K символов перед каждым вхождением символа C .
24. Даны строки S , S_1 , S_2 . Удалить из строки S каждое нечётное вхождение строки S_1 или S_2 .
25. Дана строка S , число K и символ C . Дополнить каждую последовательность не пробельных символов до длины K символами C .

Обработка слов

Решить задания использованием строк типа `string`.

Словом будем называть последовательность символов, отличных от пробела и разделённых произвольным количеством пробелов. В начале и в конце строки, состоящей из слов, может быть произвольное количество пробелов.

1. Сформировать строку из слов исходной строки, имеющих заданную длину.
2. Преобразовать в строке все слова по правилу: перенести последний символ в начало слова.
3. Сформировать строку из слов исходной строки, состоящих из символов отличных от латинских букв.
4. Преобразовать в строке все слова по правилу: если слово нечетной длины, то удалить его последний символ.
5. Удалить из строки все вхождения первого слова.
6. Сформировать строку из слов исходной строки, начинающихся и оканчивающихся одним и тем же символом.
7. Преобразовать в строке все слова по правилу: удалить из слова все последующие вхождения первого символа.
8. Сформировать строку из слов исходной строки, содержащих указанное количество заданных символов.
9. Преобразовать в строке все слова по правилу: удалить из слова все вхождения заданного символа.
10. Удалить в строке все слова максимальной длины.
11. Сформировать строку из слов исходной строки, начинающихся с заглавной и оканчивающихся строчной буквами.
12. Преобразовать в строке все слова по правилу: попарно поменять символы местами.
13. Сформировать строку из слов исходной строки, представляющих целые числа.
14. Преобразовать в строке все слова по правилу: удалить из слова все гласные буквы.
15. Удалить из строки все слова не являющиеся [идентификаторами](#).

16. Сформировать строку из слов исходной строки, поменяв в каждом порядок следования символов на обратный.
17. Преобразовать в строке все слова по правилу: заменить букву на противоположную в алфавите (*A-Z, B-Y, C-X ...*).
18. Сформировать строку из слов исходной строки, поменяв порядок слов на обратный.
19. Преобразовать в строке все слова по правилу: удалить из слова все символы не являющиеся буквами.
20. Удалить из строки все слова, представляющие числа.
21. Сформировать строку из слов исходной строки, в которых буквы упорядочены по алфавиту.
22. Преобразовать в строке все слова по правилу: дополнить слово до длины *N*, дописав в конец слова необходимое количество символов «*».
23. Сформировать строку из слов исходной строки, в которых латинские буквы чередуются с цифрами.
24. Преобразовать в строке все слова по правилу: удалить из слова средний символ. Если слово чётной длины, то ничего не удалять.
25. Удалить из строки все слова, начинающиеся с большой буквы.