**Лабораторная работа № 9**

Вариант А

1. В каждом слове текста *к-*ю букву заменить заданным символом. Если *к* больше длины слова, корректировку не выполнять.
2. В тексте каждую букву заменить ее порядковым номером в алфавите. При выводе в одной строке печатать текст с двумя пробелами между буквами, в следующей строке внизу под каждой буквой печатать ее номер.
3. В тексте после буквы Р, если она не последняя в слове, ошибочно напечата­на буква А вместо О. Внести исправления в текст.
4. В тексте после *к-*го символа вставить заданную подстроку.
5. После каждого слова текста, заканчивающегося заданной подстрокой, вста­вить указанное слово.
6. В зависимости от признака (0 или 1) в каждой строке текста удалить указан­ный символ везде, где он встречается, или вставить его после *к-* го символа.
7. Из текста удалить все символы, кроме пробелов, не являющиеся буквами. Между последовательностями подряд идущих букв оставить хотя бы один пробел.
8. Удалить из текста его часть, заключенную между двумя символами, кото­рые вводятся (например, между скобками «(» и «)» или между звездочками «\*» и т.п.).
9. Определить, сколько раз повторяется в тексте каждое слово, которое встре­чается в нем.
10. В тексте найти и напечатать *п* символов (и их количество), встречающихся наиболее часто.
11. Найти, каких букв, гласных или согласных, больше в каждом предложении текста.
12. В стихотворении найти количество слов, начинающихся и заканчивающих­ся гласной буквой.
13. Напечатать без повторения слова текста, у которых первая и последняя бук­вы совпадают.
14. В тексте найти и напечатать все слова максимальной и все слова минималь­ной длины.
15. Напечатать квитанцию об оплате телеграммы, если стоимость одного слова задана.
16. В стихотворении найти одинаковые буквы, которые встречаются во всех словах.
17. В тексте найти первую подстроку максимальной длины, не содержащую букв.
18. В тексте определить все согласные буквы, встречающиеся не более чем в двух словах.
19. Преобразовать текст так, чтобы каждое слово, не содержащее неалфавит­ных символов, начиналось с заглавной буквы.
20. Подсчитать количество содержащихся в данном тексте знаков препинания.
21. В заданном тексте найти сумму всех встречающихся цифр.
22. Из кода Java удалить все комментарии (//, /\*, /\*\*).
23. Все слова текста встречаются четное количество раз, за исключением одно­го. Определить это слово. При сравнении слов регистр не учитывать.
24. Определить сумму всех целых чисел, встречающихся в заданном тексте.
25. Из текста удалить все лишние пробелы, если они разделяют два различных знака препинания, и если рядом с ними находится еще один пробел.
26. Строка состоит из упорядоченных чисел от 0 до 100000, записанных под­ряд без пробелов. Определить, что будет подстрокой от позиции *n* до *m*.
27. Определить количество вхождений заданного слова в текст, игнорируя ре­гистр символов и считая буквы «е», «ё», и «и», «й» одинаковыми.
28. Преобразовать текст так, чтобы только первые буквы каждого предложения были заглавными.
29. Заменить все одинаковые рядом стоящие в тексте символы одним символом.
30. Вывести в заданном тексте все слова, расположив их в алфавитном порядке.
31. Подсчитать, сколько слов в заданном тексте начинается с прописной буквы. 32. Подсчитать, сколько раз заданное слово входит в текст.

Вариант B

Создать программу обработки текста учебника по программированию с ис­пользованием классов: *Символ*, *Слово*, *Предложение*, *Абзац*, *Лексема*, *Листинг*, *Знак препинания* и др. Во всех задачах с формированием текста заменять табу­ляции и последовательности пробелов одним пробелом.

Предварительно текст следует разобрать на составные части, выполнить одно из перечисленных ниже заданий и вывести полученный результат.

1. Найти наибольшее количество предложений текста, в которых есть одина­ковые слова.
2. Вывести все предложения заданного текста в порядке возрастания количе­ства слов в каждом из них.
3. Найти такое слово в первом предложении, которого нет ни в одном из остальных предложений.
4. Во всех вопросительных предложениях текста найти и напечатать без по­вторений слова заданной длины.
5. В каждом предложении текста поменять местами первое слово с послед­ним, не изменяя длины предложения.
6. Напечатать слова текста в алфавитном порядке по первой букве. Слова, на­чинающиеся с новой буквы, печатать с красной строки.
7. Рассортировать слова текста по возрастанию доли гласных букв (отноше­ние количества гласных к общему количеству букв в слове).
8. Слова текста, начинающиеся с гласных букв, рассортировать в алфавитном порядке по первой согласной букве слова.
9. Все слова текста рассортировать по возрастанию количества заданной буквы в слове. Слова с одинаковым количеством расположить в алфавитном порядке.
10. Существует текст и список слов. Для каждого слова из заданного списка найти, сколько раз оно встречается в каждом предложении, и рассортиро­вать слова по убыванию общего количества вхождений.
11. В каждом предложении текста исключить подстроку максимальной длины, начинающуюся и заканчивающуюся заданными символами.
12. Из текста удалить все слова заданной длины, начинающиеся на согласную букву.
13. Отсортировать слова в тексте по убыванию количества вхождений заданно­го символа, а в случае равенства — по алфавиту.
14. В заданном тексте найти подстроку максимальной длины, являющуюся па­линдромом, т.е. читающуюся слева направо и справа налево одинаково.
15. Преобразовать каждое слово в тексте, удалив из него все следующие (пре­дыдущие) вхождения первой (последней) буквы этого слова.
16. В некотором предложении текста слова заданной длины заменить указан­ной подстрокой, длина которой может не совпадать с длиной слова.

Вариант С

1. Проверить, является ли строка сильным паролем. Пароль считается сильным, если его длина больше либо равна 10 символам, он содержит как минимум одну цифру, одну букву в верхнем и одну букву в нижнем регистре. Пароль может содержать только латинские буквы и/или цифры, а также символ «\_».
2. Текст из *п*2 символов шифруется по следующему правилу:

* все символы текста записываются в квадратную таблицу размерности *п* в порядке слева направо, сверху вниз;
* таблица поворачивается на 90° по часовой стрелке;
* 1-я строка таблицы меняется местами с последней, 2-я — с предпо­следней и т.д.;
* 1-й столбец таблицы меняется местами со 2-м, 3-й — с 4-м и т.д.;
* зашифрованный текст получается в результате обхода результирующей таблицы по спирали по часовой стрелке, начиная с левого верхнего угла.

Зашифровать текст по указанному правилу.

1. Исключить из текста подстроку максимальной длины, начинающуюся и за­канчивающуюся одним и тем же символом.
2. Вычеркнуть из текста минимальное количество предложений так, чтобы у любых двух оставшихся предложений было хотя бы одно общее слово.
3. Осуществить сжатие английского текста, заменив каждую группу из двух или более рядом стоящих символов на один символ, за которым следует количество его вхождений в группу. К примеру, строка hellowoooorld долж­на сжиматься в hel2owo4rld.
4. Распаковать текст, сжатый по правилу из предыдущего задания.
5. Определить, удовлетворяет ли имя файла маске. Маска может содержать символы «?» (произвольный символ) и «\*» (произвольное количество про­извольных символов).
6. Буквенная запись телефонных номеров основана на том, что каждой цифре соответствует несколько английских букв: 2 — ABC, 3 — DEF, 4 — GHI, 5 — JKL, 6 — MNO, 7 — PQRS, 8 — TUV, 9 — WXYZ. Написать програм­му, которая находит в заданном телефонном номере подстроку максималь­ной длины, соответствующую слову из словаря.
7. Осуществить форматирование заданного текста с выравниванием по лево­му краю. Программа должна разбивать текст на строки с длиной, не пре­восходящей заданного количества символов. Если очередное слово не по­мещается в текущей строке, его необходимо переносить на следующую.
8. Изменить программу из предыдущего примера так, чтобы она осуществля­ла форматирование с выравниванием по обоим краям. Для этого добавить дополнительные пробелы между словами.
9. Добавить к программе из предыдущего примера возможность переноса слов по слогам. Предполагается, что есть доступ к словарю, в котором для каждого слова указано, как оно разбивается на слоги.
10. Пусть текст содержит миллион символов, необходимо сформировать из них строку путем конкатенации. Определить время работы кода. Ускорить процесс, используя класс **StringBuilder**.
11. Алгоритм Барроуза-Уиллера для сжатия текстов основывается на преобра­зовании Барроуза-Уиллера. Оно производится следующим образом: для слова рассматриваются все его циклические сдвиги, которые затем сорти­руются в алфавитном порядке, после чего формируется слово из последних символов отсортированных циклических сдвигов. К примеру, для слова JAVA циклические сдвиги — это JAVA, AVAJ, VAJA, AJAV. После сорти­ровки по алфавиту получим AJAV, AVAJ, JAVA, VAJA. Значит, результат преобразования — слово VJAA. Реализовать программно преобразование Барроуза-Уиллера для данного слова.
12. Восстановить слово по его преобразованию Барроуза-Уиллера. К примеру, полу­чив на вход VJAA, в результате работы программа должна выдать слово JAVA.
13. В Java код добавить корректные *getter* и *setter*-методы для всех полей дан­ного класса при их отсутствии.
14. В тексте нет слов, начинающихся одинаковыми буквами. Напечатать слова текста в таком порядке, чтобы последняя буква каждого слова совпадала с первой буквой следующего слова. Если все слова нельзя напечатать в таком порядке, найти такую цепочку, состоящую из наибольшего количества слов.
15. Текст шифруется по следующему правилу: из исходного текста выбирается 1, 4, 7, 10-й и т.д. (до конца текста) символы, затем 2, 5, 8, 11-й и т.д. (до конца текста) символы, затем 3, 6, 9, 12-й и т.д. Зашифровать заданный текст.
16. В предложении из *n* слов первое слово поставить на место второго, второе — на место третьего и т.д., (*п-* 1)-е слово — на место *п-* го, *п-*е слово поставить на место первого. В исходном и преобразованном предложениях между слова­ми должны быть или один пробел, или знак препинания и один пробел.
17. Все слова текста рассортировать в порядке убывания их длин, при этом все слова одинаковой длины рассортировать в порядке возрастания в них коли­чества гласных букв.