

- 1) Словом назовем всякую последовательность букв латинского алфавита длины не более 20. Предложением – всякую последовательность не более 30 слов, разделенных одним или несколькими пробелами, оканчивающиеся символом '.', '!' или '?'. Написать программу, читающую заданное предложение, контролирующую его правильность и выполняющую функцию, которая: определяет, сколько раз каждое слово встречается в предложении.
- 2) Дан указатель на список. Удалить первый элемент списка.
- 3) Напишите программу поиска нужного слова в словаре, в котором слова не упорядочены.
- 4) Вводится предложение – набор символов, заканчивающийся точкой (из файла или с клавиатуры по желанию пользователя). Необходимо разбить его на слова (последовательности букв латинского алфавита произвольной длины, все прочие символы считаются разделителями) и сформировать односвязный список, информационное поле элементов которого содержит ссылку на очередное слово. Написать программу, которая: упорядочивает слова в списке по алфавиту, затем распечатывает список в прямом и обратном порядке.
- 5) Дан указатель на дерево. Определите высоту дерева.
- 6) Удалить из данного графа наибольшее множество ребер, чтобы оставшийся граф не распался на большее число компонент связности (алгоритм Прима выделения основного леса).
- 7) Файл содержит некоторый набор символов. Назовем словом последовательность латинских букв, разделителями являются все остальные символы. Написать программу которая: во всех словах, имеющих окончание ing, заменяет на ed.
- 8) Бинарное дерево – это динамическая структура данных. Каждый узел задается в программе переменной типа структура, имеющей три части: LS(левый сын), Info(информационная составляющая) и RS(правый сын). Пусть поле Info представляет из себя строку символов длиной не более 10. Требуется написать программу, которая считывает этот файл, формирует в памяти дерево и вызывает функцию, которая выводит на экран значения поля Info узлов дерева. При этом узлы дерева появляются на экране в порядке соответствующем:
 - а) прямому обходу; б) обратному обходу; в) симметричному обходу.