# Работа с файлами и строковыми данными

Составить программу, анализирующую или обрабатывающую файл Input.txt (файл создать самим с помощью редактора NotePad или др., он должен содержать несколько строк, оканчивающихся символом перевода строки).

1. Заменить все последовательности более чем из одного пробела на один пробел, за исключением тех случаев, когда искомые комбинации находятся внутри кавычек или апострофов.
2. Заменить все строчные буквы, следующие после точки или комбинации «точка и пробел» на заглавные буквы.

Обработанный текст сохранить в выходной файл.

**Методические указания:**

Для решения поставленной задачи вам необходимо самостоятельно изучить функции работы с файлами по их описанию в библиотеке MSDN или другой справочной информации.

 В C существует 2 группы функций для работы с файлами:

* верхнего уровня (форматированный доступ к файлам), основные из них:

**fopen, fread, fwrite, fscanf, fprinf, fgetc, fgets, fclose, feof**

* нижнего уровня (неформатированный доступ к файлам), основные из них:

**open, read, write, close, eof**

*Особенность MS Visual C:* к идентификаторам этих функций приписывают вначале знак \_ (например, \_open). Возможно их написание как со знаком \_, так и без него. В библиотеке MSDN в поиске необходимо добавлять этот знак, т.е. искать сочетания \_open, \_read и т.д.

 Для подсчёта количества символов в файле необходимо считывать его содержимое посимвольно, пока файл не будет прочитан полностью (что можно определить с помощью функции eof() или feof()). Другой вариант заключается в использовании функции read, однако следует помнить, что ваша программа должна быть работоспособной для любого сколь угодно большого файла. Эти действия необходимо проделать как для текстового режима работы, так и для бинарного режима. Существует также третий метод определения длины файла, основанный на использовании функций seek() и tell().

 При использовании функций read, write, fread, fwrite не забывайте корректно выделять (динамически или статически) и освобождать память, в которую функции будут записывать содержимое файла или откуда они будут считывать записываемые значения.

 Анализ содержимого файла (поиск заданных слов) можно проводить как с помощью посимвольного сравнения (в цикле), так и с помощью процедур работы со строками.

 Для хранения полученных данных о количестве найденных слов удобно использовать массивы.