

# Строки символов

## Часть 2



# Определение размера массива строк

Если количество строк неизвестно и строки в файле

- `realloc()` использовать не рекомендуется (трудности с проверками)
- Можно экономить время и всю работу вести в памяти
- Можно экономить память, но потратить лишнее время на работу с файлом



# Определение размера массива строк

## Путь экономии времени

- Определяем количество байтов (символов) в файле
- Выделяем память под весь текст
- Считываем весь текст в большой массив (`fread()`)
- Проходим по массиву и считаем символы перевода строки (абзацы)
- Выделяем память для массива строк по количеству абзацев
- Для каждого фрагмента текста (абзаца) считаем количество символов в абзаце, выделяем память для элемента массива строк и заполняем посимвольно элемент массива строк.

Пример `lect-09-01.c`



# Определение размера массива строк

## Путь экономии памяти (проще, но дольше)

- Определяем количество строк в файле (пока `fgets( ) != NULL`)
- Выделяем память для массива строк по счетчику строк
- Для каждой прочитанной строки определяем реальную длину, выделяем память и формируем элемент массива (`strcpy( )`).

Пример `lect-09-02.c`



# Дополнительные полезные функции

- `sscanf()` – делит строку, указанную в первом аргументе, на переменные, которые задаются строкой формата (`<stdio.h>`).
- `atoi()`, `atof()` – преобразуют набор символов-цифр в числа (`<stdlib.h>`).
- `sprintf()` – создает строку из значений переменных представляемых в соответствии с форматом (`<stdio.h>`).

см. пример `lect-09-03.c`

- `isupper()`, `islower()` – проверяют регистр символов (не 0, если «да», `<ctype.h>`).
- `toupper()`, `tolower()` – преобразуют регистр символов (`<ctype.h>`).
- `strstr()` - определяет позицию (адрес) первого вхождения подстроки в строку (`<string.h>`).
- `strcat()` - дописывает новую строку в конец строки («склейка» строк, `<string.h>`).



# Комплексная задача

В произвольной строке, состоящей из слов с символами-разделителями, отсортировать слова в прямом лексикографическом порядке\*. После перестановки слов последовательности символов-разделителей остаются на прежних позициях\*\*.

\* в соответствии с кодировочной таблицей.

\*\* то, что было после N-го слова, остается после N-го слова, хотя само слово меняется.

## Пример

Исходная строка:

**ZZ Top 2000 blues... It is the beatyful music!**

Результат обработки:

**2000 It Top ZZ... beatyful blues is music the!**



# Постановка задачи (уточнения и доп. данные)

- Разделителями считается всё, что не буква и не цифра
- Строка не начинается с разделителя
- Строка может заканчиваться как буквой/цифрой, так и разделителем (разделителями)
- Слово — последовательность букв и цифр между разделителями (или с начала строки до первого разделителя, или от последнего разделителя до последней буквы/цифры в строке)
- Из слов и последовательностей разделителей формируются массивы
- Длина строки (общее количество символов в строке) заранее не известна, но не может превышать 256 символов
- Количество слов и последовательностей символов-разделителей в строке заранее не известно
- Строка может состоять из одного слова, но не может состоять только из разделителей (см. выше)
- Для упрощения работы с данными (промежуточного хранения) используются файлы
- Для ввода и вывода используются стандартные устройства (клавиатура и экран)
- Используются только набор символов латиницы и стандартные символы клавиатуры (символы ASCII).



# Порядок решения задачи (алгоритм)

- Прочитать исходную строку
- Обработать строку
  - Разделить строку на массивы слов и символов-разделителей
    - ◆ Читать символы пока не появится разделитель, сохранить полученное слово в новую строку в файле слов
    - ◆ Читать символы, пока не появится буква, сохранить последовательность символов-разделителей в новую строку в файле разделителей
    - ◆ Повторять два предыдущих действия, пока не кончится текст
  - Сформировать массив строк с разделителями из файла разделителей
  - Сформировать массив слов из файла слов
  - Отсортировать массив слов
  - Составить строку из элементов массива слов и массива строк с разделителями
- Вывести результат обработки (полученную строку).

