

## Блок-8. Машины (обязательные) по теме «ССЫЛОЧНЫЙ ТИП»

(внимание, во всех задачах этого блока обязательно подключать модуль **heaptrc**)

### Задача-1 «Выборочное копирование строк текстового файла»

Дан внешний *текстовый* файл с именем **file1.txt** (в состав файла могут входить как пустые, так и непустые строки).

За один просмотр этого файла сгенерировать другой текстовый файл с именем **file2.txt** по следующему *правилу*: во второй файл следует перенести только те строки первого файла, **длина которых больше единицы и которые при этом заканчиваются нечётной цифрой** (все остальные строки – не переносить). Файл с именем **file1.txt** следует самостоятельно подготовить в редакторе **Jeany** (или другом редакторе, где набираете тексты программ) и **прислать его вместе с программой** (содержимое файла должно соответствовать картинке **справа** →).

**Требования.** В решении **не использовать вспомогательных массивов, файлов**. Для сохранения символов текущей строки следует воспользоваться вспомогательным *линейным списком*, в звенья которого будут последовательно заноситься (в качестве элементов) символы этой строки. При переходе к чтению следующей строки следует опустошать (с освобождением памяти) список, полученный по результату просмотра предыдущей строки.

**Рекомендации.** Описать процедуру **create(L,Ok)**, которая строит список **L** из символов текущей строки исходного текстового файла, а через параметр **Ok** сообщает, годится ли прочитанная строка для переноса в результирующий файл. Описать процедуру **move(L)**, которая переносит элементы-символы из списка **L** – в очередную строку формируемого файла. Описать также процедуру **destroy(L)**, которая удаляет список **L** (с освобождением памяти). Остальные процедуры (если есть желание) – на усмотрение автора программы.

| Файл<br>file1.txt | Файл file2.txt |
|-------------------|----------------|
| asdf              | asdfg5         |
| asdfg5            | werty249       |
| 9                 | 45             |
| werty249          | 34567          |
| 45                | rtYu87878787   |
|                   | 445            |
|                   | rw9            |
| 345678            | wer7891        |
| 34567             |                |
| t                 |                |
| rtYu87878787      |                |
| 44                |                |
| 445               |                |
|                   |                |
| 2345a             |                |
| sdfgh             |                |
|                   |                |
| rw9               |                |
| w                 |                |
|                   |                |
| wer78911          |                |
| r4                |                |

### Задача-2 «Массив из списков»

Дана непустая последовательность слов, в каждом из которых содержится от **1** до **6** малых латинских букв; соседние слова разделены запятой, за последним словом следует точка. Требуется ввести (с клавиатуры) эту последовательность слов в память ЭВМ, преобразовав ее во внутреннее представление (см. ниже), а затем распечатать **в алфавитном порядке**: сначала все различные однобуквенные слова с указанием для каждого из них числа его вхождений в исходную последовательность, затем аналогичным образом все различные двухбуквенные слова и т.д. В качестве внутреннего представления последовательности слов использовать: **массив из 6 списков**, в **k**-ом из которых хранятся **k**-буквенные слова, упорядоченные по алфавиту. Для вставки нового элемента-слова в нужный список воспользоваться рекурсивной процедурой, написанной на семинаре от 28.11.2020 для задачи **16.24** (см. **слайды 28 и 29**). Потребуется лишь незначительные корректировки этой процедуры – с учётом того, что в каждом звене, помимо собственно слова, нужно хранить и число его вхождений в последовательность. Для печати отдельно взятого списка (любого из шести) описать соответствующую процедуру. Ответ выдавать на экран. Перед окончанием работы программы – освободить память, занятую под списки (с помощью процедуры **Destroy(L)**, описанной в **Задаче-1**).

**Итого: 2 обязательные задачи. Срок их сдачи – 19 декабря.**

## **Блок-8. Машины (дополнительные) по теме «СПИСКИ»**

**Во всех задачах - не допускать “утечки памяти” (модуль `heaptrc`) !!!**

**Обязательно перед началом решения повторите материалы семинара от 02.12.2020.**

### **задача 1 (20 очков)**

-----  
“Считалка” (см. 16.42 из задачника Пильщикова). Пояснение условия на примерах:

- 1) если  $n=10$ ,  $k=4$ , то последовательность удаления такая:  $e_4, e_8, e_2, e_7, e_3, e_{10}, e_9, e_1, e_6, e_5$
- 2) если  $n=10$ ,  $k=11$ , то последовательность удаления такая:  $e_1, e_3, e_6, e_{10}, e_8, e_9, e_5, e_2, e_4, e_7$
- 3) если  $n=10$ ,  $k=1$ , то последовательность удаления такая:  $e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9, e_{10}$

Решение задачи дать в соответствии со следующими требованиями:

- 1) Дан непустой текстовый файл (заранее самостоятельно его заготовить) с именем `friends.txt`, все строки которого – непустые. В каждой его строке записано (латинскими буквами) некоторое имя (совпадающих имен нет). В имени - *не более 15 букв*. Описать процедуру, которая по содержимому файла строит **кольцевой однонаправленный список**, элементами которого являются прочитанные имена (недостающие буквы в имени дополнить пробелами). Каждое очередное прочитанное имя добавляется в конец формируемого кольцевого списка (т.е. в первом звене - имя из 1-ой строки, в последнем звене - имя из последней строки). Последнее звено соединено ссылкой с первым. После построения списка процедура выводит на экран (в отдельных строках) имена ребят и количество человек в группе (по числу строк в текстовом файле, т.е. фактическое значение для  $n$ ).
- 2) Описать процедуру, которая по заданному кольцевому списку  $L$  и значению  $k$  (натуральному) выводит на экран имена ребят (каждое - с новой строки) в порядке их удаления из круга. Процедура работает с кольцевым списком, удаляя из него на каждом шаге очередное ненужное звено (обязательно с освобождением памяти). *Внимание:* если в круге останется не много людей, а  $k$  - велико, то не следует обходить круг многократно для поиска очередного звена, подлежащего удалению (можно сообразить как выйти из такого положения).

Текстовый файл `friends.txt`, над которым выполнялась Ваша программа, **поместите в отправляемый архив**. Просьба, для удобства проверки, имена, записанные в файле, помечать по номеру строки этого файла (и следовательно, номеру элемента в формируемом кольцевом списке), например:

1\_Ivan  
2\_Petr  
3\_Kate  
4\_Igor  
и т.п.

### **задача 2 (20 очков)**

-----  
Это задача на **двунаправленный список**. Условие самого задания сформулировано в номере 16.44. Требования к решению этой задачи:

- 1) Заранее (самостоятельно) подготовить текстовый файл с именем **t1.txt**, в строках которого могут появляться символы **#**. В файле допускаются и пустые строки.
  - 2) Программа должна по файлу **t1.txt** построить новый файл **t2.txt**. Каждая строка нового файла должна быть преобразована в соответствии с правилом, приведенным в условии упражнения **16.44**.
  - 3) Преобразование строки следует выполнять с использованием двунаправленного списка (сообразить самостоятельно, какие операции потребуется реализовать над этим списком).
  - 4) Пустые строки или строки, ставшие пустыми в процессе преобразования, переносить в новый файл запрещено.
- Внимание:* а) в новом файле не должно быть символов **#**; б) в новом файле не должно быть *пустых строк*
- 5) Прислать файлы **t1.txt** и **t2.txt** (свой вариант). В исходном файле **t1.txt** должно быть **не менее 12 строк** разного вида (среди которых - **не менее двух пустых**).

### задача 3 (20 очков)

Дан непустой текстовый файл **words.txt**, в каждой строке которого записано по одному непустому слову (пробелы в записи слов не используются). Вывести на экран (в разных строках) **все слова максимальной длины**. Решить эту задачу за один просмотр исходного файла. В решении использовать **иерархический список** (в нем формируется ответ по принципу: элементы основного списка хранят ссылки на вспомогательные списки, полученные из букв слов одинаковой /максимальной длины). Не сохранять где-либо исходный текст полностью! Хранить только список из слов максимальной длины (на текущий момент обработки файла). Если выяснится, что слова из списка не подошли (появилось более длинное слово), освободить занимаемую этими словами память и обновлять список. Поместить в отправляемый архив свой вариант файла **words.txt**.

### задача 4 (20 очков)

Решить задачу **12.35** (блока-6 части-3) в следующей постановке: если из заданного начального (*first*) города можно добраться в заданный конечный (*last*) город, то вывести все возможные пути (каждый путь печатать с новой строки) в следующем порядке: Сначала путь длины **1** (если такой есть), затем все пути длины **2** (если такие есть), и т.п., в конце – все пути длины **n** (если такие есть). Программа должна быть оттранслирована для значения **n=15**.

Самостоятельно продумать, какая структура необходима для представления данных в этой задаче. Считывать **пары чисел** (номера городов, между которыми есть дорога) следует из текстового файла **roads.txt** (каждая пара чисел должна быть задана в отдельной строке, в последней строке должна находиться пара **0 0**), ввод **числовой**. Печать результатов осуществлять в текстовый файл **results.txt**, вывод **числовой**. Вместе с программой прислать **свой вариант файла roads.txt** (в файле должно быть **не менее 20 пар чисел**), а также **картинку** (можно фото с рукописного рисунка) используемой в решении **структуры данных** для хранения всех дорог.

# И снова в бой!