

Блок-7 Машины (обязательные) по теме «ФАЙЛЫ»

Задача 1 «Строки-слова из латинских букв»

В *текстовом* файле с названием **words.txt** задана **непустая** последовательность из **непустых** слов. Слова составлены из *строчных* (маленьких) *латинских* букв. Каждое слово занимает *одну строку* этого текстового файла (т.е. каждое слово завершается “концом строки”), в каждом слове – не менее одной буквы. Необходимо проанализировать этот файл и напечатать *в алфавитном порядке*:

- 1) все **согласные** буквы, которые **входят хотя бы в одно слово**;
- 2) все **согласные** буквы, которые **не входят ни в одно слово**;
- 3) все **гласные** буквы, которые **входят в каждое слово**;
- 4) все **гласные** буквы, которые **не входят хотя бы в одно слово**;

Ответ сформировать в текстовом файле с названием **result.txt** (ответ для каждого пункта должен размещаться в отдельной строке этого текстового файла), например, так (можно и в более красивом виде, что приветствуется ☺):

```
1)  c g m p r t z
2)  b d f k r
3)  a i o u y
4)  o y
```

Не следует при выполнении этого задания вводить вспомогательных процедур и функций, запрещено использовать массивы. Использовать обязательно множества (поймите самостоятельно для чего они нужны)! Решать задачу за один просмотр текстового файла. В названии дискового файла **words.txt** все буквы – **маленькие латинские**!

Файлы **words.txt** и **result.txt** должны быть расположены в том же каталоге, что и сама программа (т.е. в текущем каталоге). *Внимание:* исходный файл **words.txt** следует набирать в обычном текстовом редакторе (например, в **Jeany**, либо в другом любимом редакторе, где набираете тексты ваших программ). Сгенерированный в результате работы вашей программы файл **result.txt** можно также просмотреть с помощью того же текстового редактора (чтобы увидеть что в итоге получилось).

При сдаче этого задания **прислать**: свой текстовый файл **words.txt** из не менее десяти непустых строк-слов (который использовался для тестирования вашей программы), а также файл **result.txt** с результатами работы вашей программы. Не забудьте, естественно, прислать и саму программу.

Задача 2 «Равенство текстовых файлов»

Написать программу, которая проверяет на равенство два *текстовых* файла с названиями **t1.txt** и **t2.txt** (для тестирования программы используйте любые свои текстовые файлы; я же буду проверять на своих файлах). Проверяемые файлы **t1.txt** и **t2.txt** должны быть расположены в том же каталоге, что и сама программа (т.е. в текущем каталоге). См. рекомендации по формированию и использованию текстов **t1.txt** и **t2.txt** в предыдущем задании.

Замечание: текстовые файлы равны, если у них одинаковая структура (количество строк) и все соответствующие строки равны по длине и по входящим в них символам.

Задача 3 «Самая длинная строка»

Написать программу, которая выводит на экран первую из самых длинных строк *текстового* файла с именем **t3.txt** (для тестирования программы используйте любой текстовый файл; я же буду проверять на своём файле). Проверяемый файл **t3.txt** должен быть расположен в том же каталоге, что и сама программа (т.е. в текущем каталоге). См. рекомендации по формированию и использованию текста **t3.txt** в первом задании.

Замечание: рекомендуется выполнить два прохода по файлу: 1-ый проход - поиск порядкового номера строки наибольшей длины (например, в переменной **k**); 2-ой проход - движение по строкам файла с помощью цикла с параметром (**k-1**) шагов и установка маркера на начало **k**-ой строки, после чего строку следует распечатать (на экран).

Задача 4 «Студенты-хорошисты»

Имеется типизированный файл с названием **course.bin** типа **course** :

```
type str = packed array[1..10] of char;  
stud = record { информация об одном студенте }  
    fn: record fam, name: str end; { фамилия, имя }  
    sex: (M,W); { пол: муж (M), жен (W) }  
    marks: array[1..5] of 2..5 { пять оценок }  
end;  
course = file of stud;
```

(**Внимание:** в программе эти типы следует описывать **без каких-либо изменений** !)

Считая, что количество элементов (студентов) в этом файле заранее неизвестно, составить программу для **чтения** данных из этого файла и **записи** в текстовый файл с названием **ans.txt** следующей информации: о каждом «ХОРОШИСТЕ» (все его/её оценки 4 и 5 и при этом есть хотя бы одна 4) в отдельной строке сообщить **фамилию, имя, пол и все полученные оценки** (пример возможной строки: 'GORYACHAYA ILONA W 5_4_5_4_5'). Список «ХОРОШИСТОВ» выдавать **упорядоченным по алфавиту** (при упорядочении учитывать только фамилии). *Замечание:* в исходном файле **course.bin** фамилии и имена записаны **большими латинскими буквами**; при сортировке фамилий **можно** учитывать тот факт, что в типе **Char** большие латинские буквы упорядочены по алфавиту.

Требования: 1) ввести логическую функцию для анализа текущей записи файла (представляет ли эта запись интерес?); 2) не использовать вспомогательных массивов для хранения записей и их сортировки; 3) построить список (упорядоченный по неубыванию) из записей, представляющих интерес; этот список формировать по ходу чтения элементов из файла **course.bin**; 4) ввести **рекурсивную** процедуру, которая вставляет в этот список новый элемент с сохранением упорядоченности (процедура должна работать корректно и для пустого списка); 5) выполнить (**в цикле**) просмотр построенного списка и вывести (в файл **ans.txt**) требуемую информацию (при выводе воспользоваться оператором присоединения **with**, чтобы обращаться к полям иерархической записи напрямую, по их именам – без выписывания точек). *Внимание:* 1) строковые значения выводите в файл за одно обращение к оператору **write**, а не посимвольно (для «красоты» используйте форматный вывод строк); 2) учтите, что в Стандарте значения перечислимых типов выводить в текстовый файл по процедуре **write** нельзя (выход см. в ранее рассмотренной на семинаре задаче 7.13а).

Предостережение: исходный файл **course.bin** не является текстовым, это обычный (типизированный) файл (т.е. файл типа **file of TE**), просмотр которого из текстового редактора не имеет большого смысла (т.к. данные в нём хранятся во внутреннем машинном/двоичном представлении).

Сам файл **course.bin** прилагаю к этому письму. Прежде чем начать решать собственно задачу, рекомендую **расшифровать содержимое этого файла** и сохранить полученную информацию в каком-нибудь вспомогательном текстовом файле (иначе вам невозможно будет тестировать правильность работы программы).

Итого: 4 обязательных задачи.

Срок сдачи 1, 2, 3-ей задач – 9 декабря. Срок сдачи 4-ой задачи – 16 декабря.

Блок-7 Машины (дополнительные) по теме «ФАЙЛЫ»

Задача 1 «Комментарии» (15 очков)

Это задача **15.60** (из задачника)

Вместе с самой программой пришлите также свой исходный текстовый файл **prog.pas** (в файле должно быть не менее 20 строк, с наличием в нём **собственно комментариев**, а также с присутствием в файле **символьных значений** '{', '}' и **строк** вида 'qw{cv}v', '{s{{d}b}' и т.п.). Пришлите также получившийся в результате обработки текстовый файл **prog1.pas**

Задача2 «Сбалансированное слияние» (15 очков)

Это задача **15.64** (из задачника)

Замечание. Все четыре файла (**a.txt** – основной; **b.txt**, **c.txt**, **d.txt** - вспомогательные) должны быть *текстовыми* со следующими условиями: *в каждой строке* текстового файла хранится *одно целое число*. То есть следует использовать *числовой ввод-вывод* при работе с текстовыми файлами (вспомните рассмотренное нами упражнение **15.51 пункты д-и**). Вместе с программой пришлите свой исходный текстовый файл **a.txt** (со своей тестовой последовательностью – из не менее 20 чисел).

Ну что ж, в бой !!!