Расчетно-графическая работа

P1 [#20]

Разработайте программу BuildIndex1, которая анализирует входной текстовый файл, разбивает его на отдельные слова и строит текстовый файл индекса в следующем формате:

```
<слово> <количество вхождений в исходном файле>
```

Записи в файле индекса должны быть отсортированы по полю <слово>.

Данные хранить в связном списке, реализованном на массиве. Для хранения слов использовать тип STRING ограниченной длины. Разумное ограничение длины слова — 30-50 символов.

Размер массива ограничен возможностями компилятора. При переполнении массива завершать программу, выдавая соответствующее сообщение.

P2 [#10]

Модифицируйте конечный автомат в программе таким образом, чтобы дефисы, находящиеся между двух букв, обрабатывались как часть слова. В слове может быть несколько дефисов: Ростов-на-Дону.

P3 [#20]

Сортировку выполнять в оперативной памяти, с помощью списка или бинарного дерева. Затем содержимое списка должно выводиться в выходной файл.

P4 [#40]

Модифицируйте программу таким образом, чтобы она не зависела от количества доступной оперативной памяти.
Сделать в программе ограничение на максимальное количество слов, которое может храниться в памяти одновременно. При достижении этого ограничения отправлять данные в выходной файл, сохраняя его упорядоченность (решить задачу слияния двух упорядоченных последовательностей, см. сортировку слиянием, лекция 10, программа RecursiveSort).
Стандартное решение: слияние двух текстовых файлов в третий.
Хорошее решение: слияние списка с текстовым файлом. **(+5 баллов)**
Очень хорошее решение: слияние дерева с текстовым файлом. **(+20 баллов)**
При этом необходимо правильно работать с памятью, не допускать ее утечек.

При выполнении РГР можно использовать все возможности языка PASCAL.

Для справки: строки (тип данных STRING)

Тип данных STRING определяет строки с максимальной длиной 255 символов. Переменная такого типа может хранить значения переменной длины.

Объявление переменных типа STRING:

```
VAR
  S1: STRING; {строка максимальной длиной 255 символов}
  S2: STRING[30]; {строка максимальной длиной 30 символов}
```

Для строк определена операция конкатенации (+):

```
S3 := S1 + S2
```

Операции сравнения (=, <>, >=, <=, >, <) сравнивают строки в лексикографическом порядке.

Обращение к отдельным буквам строки производится аналогично массивам. Индексы букв считаются с 1. Элемент строки с индексом 0 хранит длину строки.

Длину строки можно узнать функцией Length:

```
S1 := 'String';
WRITELN(Length(S1));     //6
```

Операторы READ и WRITE работают со строками. Пример: копирование INPUT в OUTPUT.

```
PROGRAM CopyStrings(INPUT, OUTPUT);
VAR
  S1: STRING[30];
BEGIN
  WHILE NOT EOF
  DO
    BEGIN
      READ(S1); {Чтение не более 30 символов до наступления EOLN}
      WRITE(S1);
      IF EOLN {Если подошли к концу строки, его нужно обработать}
      THEN
        BEGIN
          READLN;
          WRITELN
        END
    END
END.
```

