# Лабораторная работа 7

#### 7.1

Выполнить разработку программы BubbleSort из приведенных ниже разделов проекта. Выполните это в следующей последовательности:

##### a) [#20]

Разработайте план сборки программы. Запишите его в следующей форме:

| **План сборки программы Example**  Этап 1 Разраб.программа 8А = DP6 +DP8.3 + DP8.3.1=15(стр.)+14(стр.)+10(стр.) =34(стр)  Этап 2 Разраб.программа 8В = 8A + DP8.2 =34(стр.) + 18(стр.)= 52(стр.)  и т.д... |
| --- |

Таким образом, для каждого этапа укажите номера собираемых разделов проекта и их размеры. Подумайте, что каждый этап будет делать, и как вы его будете тестировать.

Для программы BubbleSort потребуется три этапа сборки.

##### b) [#20]

Сборка программы с обязательным тестированием каждого этапа.

Выполнение:

INPUT:34251  
OUTPUT:12345

INPUT:11523  
OUTPUT:11235

Разделы проекта:

DP7

PROGRAM BubbleSort(INPUT, OUTPUT);

{Сортируем первую строку INPUT в OUTPUT}

VAR

  Sorted, Ch, Ch1, Ch2: CHAR;

  F1, F2: TEXT;

BEGIN {BubbleSort}

  { Копируем INPUT в F1 }

  Sorted := 'N';

  WHILE Sorted ='N'

  DO

    BEGIN

      { Копируем F1 в F2,проверяя отсортированность

       и переставляя первые соседние символы по порядку}

      { Копируем F2 в F1 }

    END;

  { Копируем F1 в OUTPUT }

END. {BubbleSort}

DP 7.1

BEGIN { Копируем F1 в F2, проверяя отсортированность

  и переставляя первые соседнии символы по порядку}

  Sorted := 'Y';

  RESET(F1);

  REWRITE(F2);

  IF NOT EOLN(F1)

  THEN

    BEGIN

      READ(F1,Ch1);

      WHILE NOT EOLN(F1)

      DO { По крайней мере два символа остается для Ch1,Ch2 }

        BEGIN

          READ(F1, Ch2);

          { Выводим   min(Ch1,Ch2) в  F2, записывая

           отсортированные символы }

        END;

      WRITELN(F2, Ch1) { Выводим последний символ в F2 }

   END

END

DP 7.1.1

{ Выводим   min(Ch1,Ch2) в  F2, записывая отсортированные символы }

IF Ch1 <= Ch2

THEN

 BEGIN

WRITE(F2, Ch1);

Ch1 := Ch2

END

ELSE

BEGIN

WRITE(F2, Ch2);

Sorted := 'N'

END

DP 7.2

BEGIN { Копируем INPUT в F1 }

  REWRITE(F1);

  WHILE NOT EOLN

  DO

    BEGIN

     READ(Ch);

      WRITE(F1, Ch);

    END;

  WRITELN(F1)

END;

DP 7.4

BEGIN { Копируем F2 в F1 }

  .................

  (аналогично DP 7.2)

END

DP 7.5

BEGIN { Копируем F1 в OUTPUT }

  .................

  (аналогично DP 7.2)

END

#### 7.2 [#10]

Приведите содержимое файла F1 на каждом проходе BubbleSort для следующих входных данных: DBCA

Внесите минимальные изменения в программу, чтобы продемонстрировать это.

#### 7.3 [#10]

Переделайте BubbleSort в программу BubbleSortDown, которая сортирует файл в порядке убывания.

#### 7.4

Переделайте BubbleSort в программу, обрабатывающую входные файлы с несколькими строками.

##### a) [#10]

Весь файл сортируется вместе и печатается как одна строка.

(Программа BubbleSortMLA)

##### b) [#15]

Каждая строка сортируется отдельно и печатается как отдельная строка.

(Программа BubbleSortMLB)