# Лабораторная работа 24

#### 24.1 [#20]

Соберите программу TreeSort. Для неотсортированной входной строки из 10 символов нарисуйте в тетради построенное программой бинарное дерево.

#### 24.2 [#30]

Сравните трудоемкость алгоритмов InsertSort2 и TreeSort в терминах количества просматриваемых пред вставкой элементов соответствующих связанных структур.

DP 3

PROGRAM TreeSort(INPUT, OUTPUT);

TYPE

Tree = ^NodeType;

NodeType = RECORD

Ch: CHAR

LLink, RLink: Tree;

END;

VAR

Root: Tree;

Ch: CHAR;

BEGIN {TreeSort}

Root := NIL;

WHILE NOT EOLN

DO

DEGIN

READ(Ch);

Insert(Root, Ch)

END;

PrintTree(Root)

END. {TreeSort}

DP 3.1

PROCEDURE Insert(Ptr:Tree, Data: CHAR);

BEGIN {Insert}

IF Ptr = NIL

THEN

BEGIN {Создаем лист со значением Data}

NEW(Ptr);

Ptr^.LLink := NIL;

Ptr^.RLink := NIL;

END

ELSE

IF Ptr^.Ch > Data

THEN

Insert(Ptr.LLink, Data)

ELSE

Insert(Ptr.RLink. Data)

END; {Insert}

DP 3.2

PROCEDURE PrintTree(Ptr: Tree);

BEGIN {PrintTree}

IF Ptr <> NIL

THEN {Печатает поддерево слева, вершину, поддерево справа}

BEGIN

PrintTree(Ptr^.LLink);

WRITE(Ch);

PrintTree(Ptr^.RLink);

END;

WRITELN

END; {PrintTree}