

**ФЕДЕРАЛЬНОЕ АГЕНТСТВО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ДИМИТРОВГРАДСКИЙ ИНСТИТУТ ТЕХНОЛОГИЙ, УПРАВЛЕНИЯ И ДИЗАЙНА
"КАФЕДРА ВЫСШЕЙ МАТЕМАТИКИ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ"**

**Лабораторная работа №3
по курсу "Алгоритмы и Структуры данных"
на тему: "Деревья"**

Выполнил студент группы ВТ-21:

Потеренко А.Г.

Проверил преподаватель:

Мингалиев Р.Ш.

Порядок работы.

1. Анализ индивидуального задания и разработка способов представления объектов задачи в памяти, методов доступа к ним.
2. Разработка программы на языке Паскаль.
3. Разработка контрольных примеров.
4. Отладка программ.
5. Составление отчета.

Содержание отчета.

	Стр.
1. Текст постановки задачи.....	3
2. Изложение способов представления объектов задачи в памяти и методов доступа к ним.....	3
3. Алгоритм.....	4-5
4. Описание и обоснование контрольных примеров.....	5
5. Текст программы с комментариями (в виде приложения).....	6-10

1. Текст постановки задачи.

Построить дерево поиска с приоритетом, реализовать функции вставки, удаления и поиска заданного узла.

2. Изложение способов представления объектов задачи в памяти и методов доступа к ним.

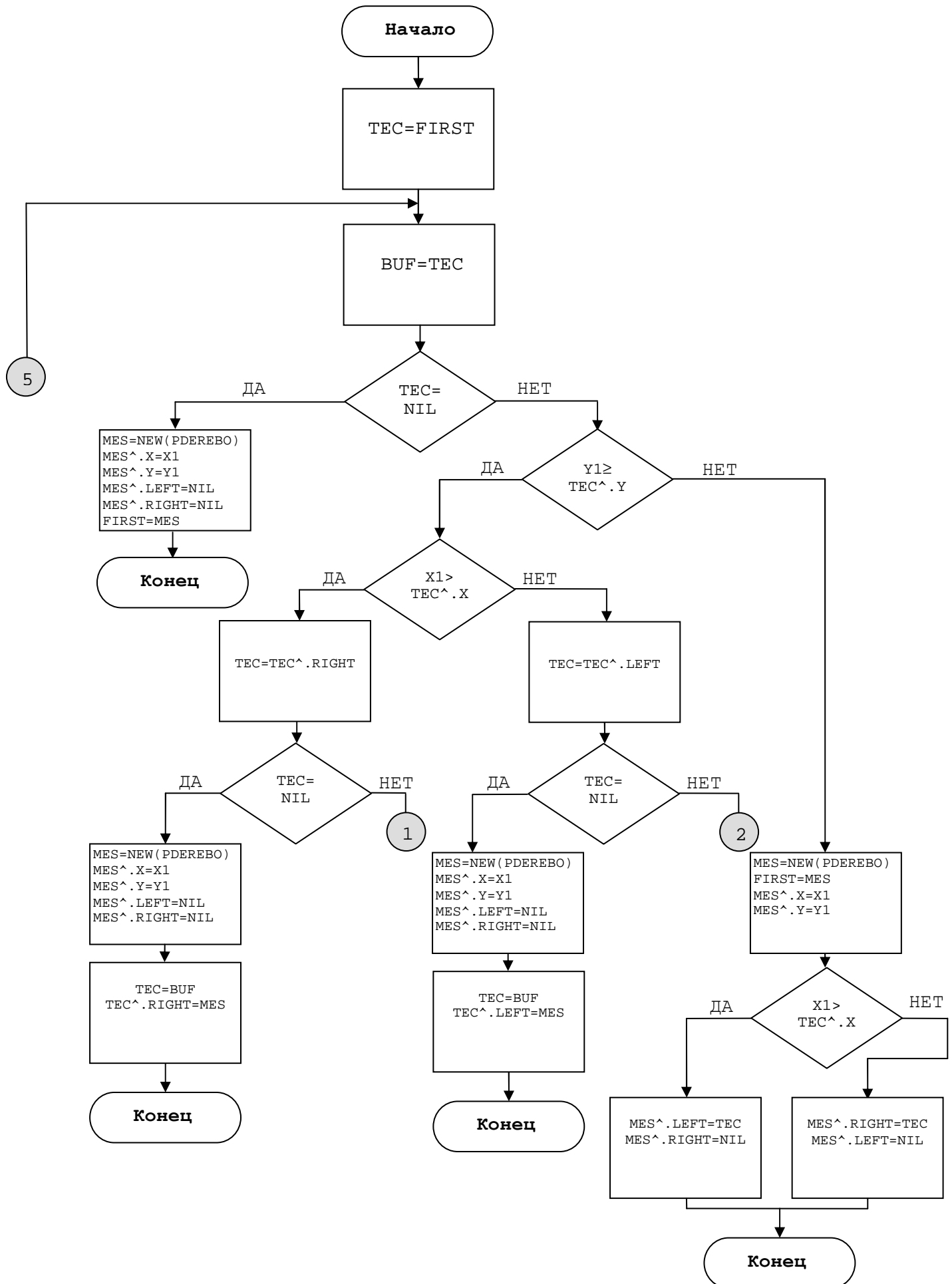
Дерево приоритетного поиска имеет два ключа: x и y , то есть представляет собой множество точек на плоскости, в двумерном декартовом пространстве, поэтому такие деревья называются так же декартовыми деревьями. Для них характерны следующие инварианты, справедливые для каждой вершины p :

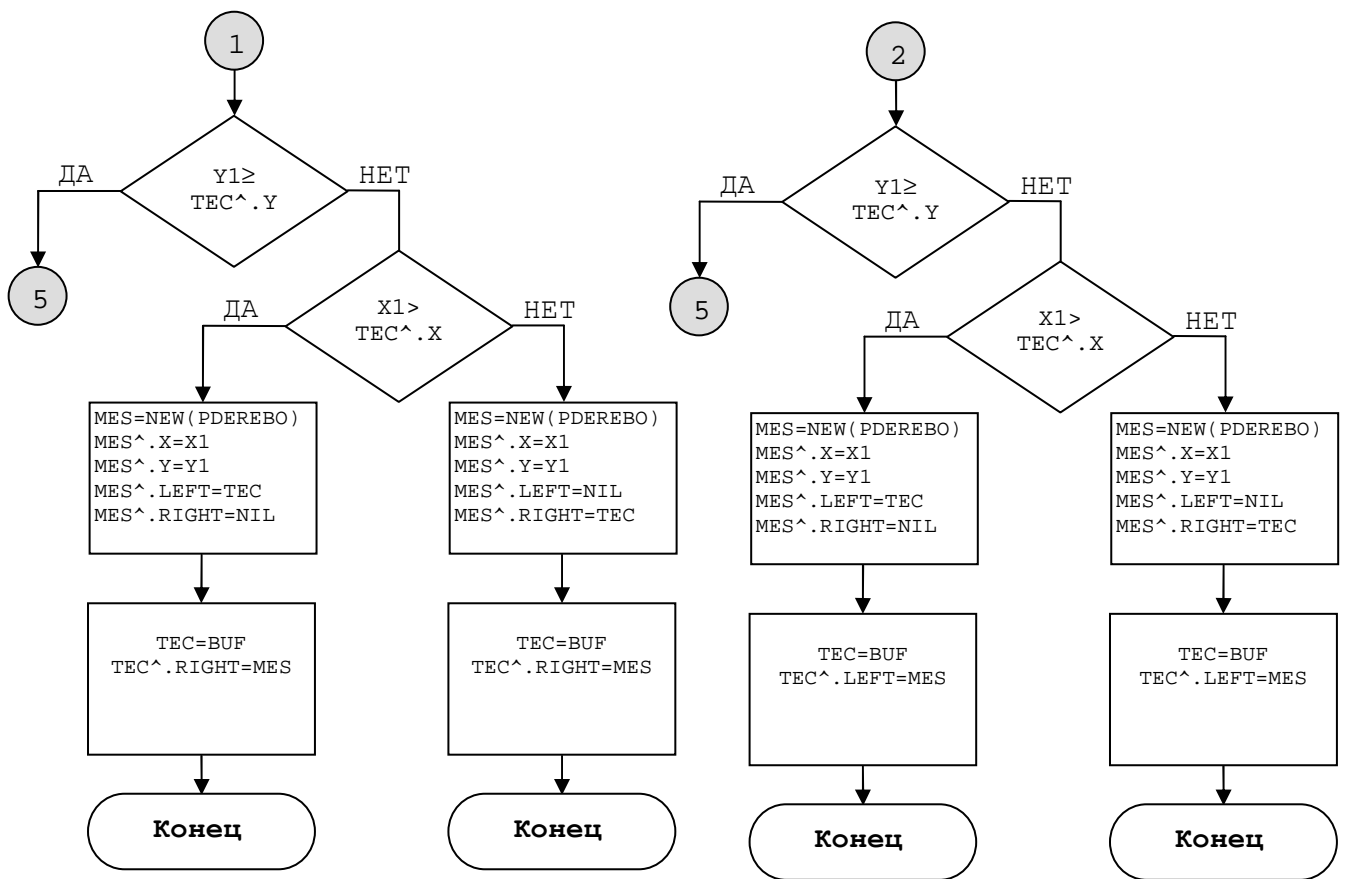
$$P.left \neq nil \rightarrow (p.left.x < p.x) \& (p.y \leq p.left.y)$$

$$P.right \neq nil \rightarrow (p.x < p.right.x) \& (p.y \leq p.right.y)$$

Из-за невозможности сохранения структуры дерева при удалении вершины, нам приходится сохранять координаты узлов дерева в динамическом массиве записей с полями обоих ключей и перестраивать дерево заново.

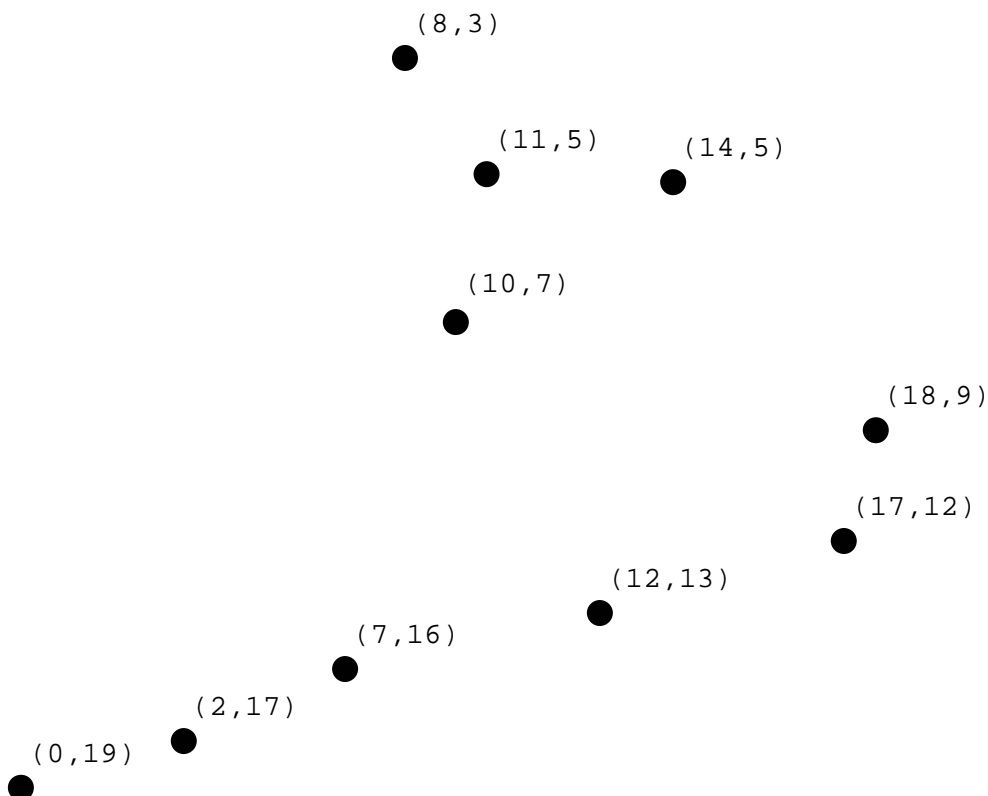
3. Алгоритм решения данной задачи: процедура вставки (X1,Y1).





4. Описание и обоснование контрольных примеров.

Программа генерирует координаты узлов из диапазона $[0, 20]$. При этом она генерирует их так, чтобы ключ x заданного узла отличался от предыдущего. Пример одной такой регенерации программы: $(8, 3)$, $(18, 9)$, $(12, 13)$, $(11, 5)$, $(10, 7)$, $(17, 12)$, $(14, 5)$, $(7, 16)$, $(2, 17)$, $(0, 19)$. Вид дерева в графическом режиме:



5. Текст программы с комментариями (в виде приложения).

```
{*****}
{*****LABORATORNAA RABOTA #3 PO ALGORITMAM *****}
{*****}
{$M 65000,0,65000}
PROGRAM LAB_3;
USES CRT,GRAPH;
CONST N=10;
      MUST=3;
TYPE PDEREBO=^DEREBO;
      DEREBO=RECORD
          X:BYTE;
          Y:BYTE;
          LEFT:PDEREBO;
          RIGHT:PDEREBO;
      END;
      DEREBO MAS=RECORD
          X:BYTE;
          Y:BYTE;
      END;
      MAS=ARRAY[1..N] OF DEREBO MAS;
VAR I,K,J,Y1,X1,LIP:INTEGER;TEC,BUF,FIRST,MES:PDEREBO;
    GD,GM:INTEGER;
    FLAG1:BOOLEAN;
    PMAS:^MAS;
    P1:WORD;
PROCEDURE BSTAVKA;
LABEL FOKS;
VAR I:INTEGER;
BEGIN
    TEC:=FIRST;
    WHILE TRUE DO
        BEGIN
            BUF:=TEC;
            IF TEC<>NIL THEN
                IF Y1>=TEC^.Y THEN
                    IF X1>TEC^.X THEN
                        BEGIN
                            TEC:=TEC^.RIGHT;
                            IF TEC=NIL THEN
                                BEGIN
                                    MES:=NEW(PDEREBO);
                                    MES^.X:=X1;
                                    MES^.Y:=Y1;
                                    MES^.LEFT:=NIL;
                                    MES^.RIGHT:=NIL;
                                    TEC:=BUF;
                                    TEC^.RIGHT:=MES;
                                    EXIT;
                                END
                            ELSE
                                IF Y1>=TEC^.Y THEN
                                    GOTO FOKS
                                ELSE
                                    IF X1>TEC^.X THEN
                                        BEGIN
                                            MES:=NEW(PDEREBO);
                                            MES^.X:=X1;
                                            MES^.Y:=Y1;
                                            MES^.LEFT:=TEC;
                                            MES^.RIGHT:=NIL;
```

```

TEC:=BUF;
TEC^.RIGHT:=MES;
EXIT;
END
ELSE
BEGIN
MES:=NEW(PDEREBO);
MES^.X:=X1;
MES^.Y:=Y1;
MES^.LEFT:=NIL;
MES^.RIGHT:=TEC;
TEC:=BUF;
TEC^.RIGHT:=MES;
EXIT
END;

END
ELSE
BEGIN
TEC:=TEC^.LEFT;
IF TEC=NIL THEN
BEGIN
MES:=NEW(PDEREBO);
MES^.X:=X1;
MES^.Y:=Y1;
MES^.LEFT:=NIL;
MES^.RIGHT:=NIL;
TEC:=BUF;
TEC^.LEFT:=MES;
EXIT;
END
ELSE
IF Y1>=TEC^.Y THEN
GOTO FOKS
ELSE
IF X1>TEC^.X THEN
BEGIN
MES:=NEW(PDEREBO);
MES^.X:=X1;
MES^.Y:=Y1;
MES^.LEFT:=TEC;
MES^.RIGHT:=NIL;
TEC:=BUF;
TEC^.LEFT:=MES;
EXIT;
END
ELSE
BEGIN
MES:=NEW(PDEREBO);
MES^.X:=X1;
MES^.Y:=Y1;
MES^.LEFT:=NIL;
MES^.RIGHT:=TEC;
TEC:=BUF;
TEC^.LEFT:=MES;
EXIT
END;

END
END
ELSE
BEGIN
MES:=NEW(PDEREBO);
FIRST:=MES;

```

```

MES^.X:=X1;
MES^.Y:=Y1;
IF X1>TEC^.X THEN
  BEGIN
    MES^.LEFT:=TEC;
    MES^.RIGHT:=NIL;
    EXIT;
  END
ELSE
  BEGIN
    MES^.RIGHT:=TEC;
    MES^.LEFT:=NIL;
    EXIT;
  END
END
ELSE
  BEGIN
    MES:=NEW(PDEREBO);
    MES^.X:=X1;
    MES^.Y:=Y1;
    MES^.LEFT:=NIL;
    MES^.RIGHT:=NIL;
    FIRST:=MES;
    EXIT;
  END;
FOKS:
END; {END WHILE}
END; {END BSTAVKA}
{*****}
PROCEDURE PROSMOTR(TEC:PDEREBO);
VAR X2,Y2,K1,K2:INTEGER;S,M1,M2:STRING;
BEGIN
  IF TEC=NIL THEN EXIT
  ELSE BEGIN
    X2:=TEC^.X;
    Y2:=TEC^.Y;
    X2:=X2*20;
    Y2:=Y2*20;
    K1:=X2;
    K2:=Y2;
    CIRCLE(X2,Y2,2);
    X2:=ROUND(X2/20);
    Y2:=ROUND(Y2/20);
    STR(X2,M1);
    STR(Y2,M2);
    S:=M1+', '+M2;
    X2:=K1;
    Y2:=K2;
    OUTTEXTXY(X2+3,Y2,S);
    PROSMOTR(TEC^.LEFT);
    PROSMOTR(TEC^.RIGHT);
  END;
END;
{*****}
PROCEDURE PROVERKA(TEC:PDEREBO);
BEGIN
  IF TEC=NIL THEN EXIT
  ELSE
    BEGIN
      IF TEC^.X=X1 THEN BEGIN FLAG1:=TRUE; EXIT;END;
      PROVERKA(TEC^.LEFT);
    END
  END

```



```

    PROVERKA(TEC^.RIGHT);
END;
END;
{*****}
PROCEDURE OSBOBOGDENIE(TEC:PDEREBO);
BEGIN
    IF TEC^.LEFT<>NIL THEN OSBOBOGDENIE(TEC^.LEFT);
    IF TEC^.RIGHT<>NIL THEN OSBOBOGDENIE(TEC^.RIGHT);
    DISPOSE(TEC);
    TEC:=NIL;
END;
{*****}
PROCEDURE KILLER(TEC:PDEREBO);
BEGIN
    IF TEC=NIL THEN EXIT
    ELSE
        BEGIN
            IF TEC^.X<>X1 THEN
                BEGIN
                    K:=K+1;
                    PMAS^[K].X:=TEC^.X;
                    PMAS^[K].Y:=TEC^.Y;
                END;
            KILLER(TEC^.LEFT);
            KILLER(TEC^.RIGHT);
        END;
    END;
END;
{*****}
PROCEDURE PERESTROIKA;
BEGIN
    FOR I:=1 TO K DO
        BEGIN
            X1:=PMAS^[I].X;
            Y1:=PMAS^[I].Y;
            BSTAVKA;
        END;
    END;
{*****TELO PROGRAMMA*****}
BEGIN
    CLRSCR;
    P1:=MEMAVAIL;
    RANDOMIZE;
    J:=0;
    REPEAT
        X1:=RANDOM(20);
        Y1:=RANDOM(20);
        FLAG1:=FALSE;
        PROVERKA(FIRST);
        IF FLAG1=FALSE THEN
            BEGIN
                J:=J+1;
                BSTAVKA;
                WRITELN('X-',J,': ',X1);
                WRITELN('Y-',J,': ',Y1);
            END;
    UNTIL J=N;
{*****}
    READKEY;
    GD:=DETECT;
    INITGRAPH(GD,GM,' ');
    PROSMOTR(FIRST);

```

```

READKEY;
{ ***** }
CLOSEGRAPH;
FOR LIP:=1 TO MUST DO
  BEGIN
    K:=0;
    WRITE('BBEDITE UD. USEL:');
    READ(X1);
    NEW(PMAS);
    KILLER(FIRST);
    OSBOBOGDENIE(FIRST);
    FIRST:=NIL;
    PERESTROIKA;
    DISPOSE(PMAS);
    PMAS:=NIL;
    { ***** }
    GD:=DETECT;
    INITGRAPH(GD,GM,' ');
    PROSMOTR(FIRST);
    READKEY;
    CLOSEGRAPH;
  END;
  OSBOBOGDENIE(FIRST);
  WRITELN('PAMAT DO:',P1);
  WRITELN('PAMAT POSLE:',MEMAVAIL);
  READKEY;
  { ***** }
END.

```