

Краткое описание

Создаётся дерево, в нижних узлах которого находятся некоторые значения. В зависимости от того на каком уровне находится исполнитель он выбирает наибольшее или наименьшее значение из значений которые находятся в узлах предыдущего уровня. Если номер уровня чётный то исполнитель выбирает наименьшее значение, иначе наибольшее. После этого исполнитель поднимается на уровень выше и проделывает ту же операцию до тех пор пока не дойдёт до верхнего уровня дерева.

```
program new17view;
uses crt,graph;
type
  shablon=^address;
  address=record
    a:integer;{переменная в которой хранятся ЗНАЧЕНИЯ узлов дерева}
    ad_l:shablon;{адреса левых узлов дерева}
    ad_r:shablon;{адреса правых узлов дерева}
  end;

var
  ad:shablon;
  i,d:integer;
procedure tree(uk:shablon;n:integer);{Процедура создания дерева}
var
  uk1:shablon;
begin
  if n<6 then
  begin
    if n<>5 then uk^.a:=0 else{Сначала создаётся второй уровень на
котором}
    begin
      uk^.a:=random(100)+1; {находятся два узла.Затем так как процедура}
      {рекурсивная она для каждого узла делает
два}
      {write('проба=',uk^.a); {вызова самой себя и т.д. пока не
создастся }
      readkey;} {заданное кол-во уровней}
    end;
    {left}
    if n<>5 then
    begin
      uk1:=uk;
      new(uk1^.ad_l);
      uk1:=uk1^.ad_l;
      tree(uk1,n+1);
      {righth}
      new(uk^.ad_r);
      uk:=uk^.ad_r;
      tree(uk,n+1);
    end;
  end;
end;
function change(uk:shablon;n:integer):integer;{Функция, которая
осуществляет}
```

```

var l,r,s,b:integer; {метод "минимакса"}
begin
  if n<6 then
    begin
      if n=5 then change:=uk^.a {к переменной "change" присваивается
ЗНАЧЕНИЕ узла нижней ступени}
      else
        begin
          l:=change(uk^.ad_l,n+1);{переменная "l" получает ЗНАЧЕНИЕ полученное
с левой ветки нижнего уровня}
          r:=change(uk^.ad_r,n+1);{переменная "r" получает ЗНАЧЕНИЕ полученное
с правой ветки нижнего уровня}
          if n mod 2=0 then {проверяется чётный или не четный уровень дерева}
            if l>=r then s:=r else s:=l {Выбирается меньшее ЗНАЧЕНИЕ узла
дерева}
            else if l>r then s:=l else s:=r;{Выбирается большее ЗНАЧЕНИЕ узла
дерева}
          change:=s;
          writeln('s=',s);readkey;
        end;
    end;
  end;
begin
  {randomize;}
  clrscr;
  new(ad);i:=0;
  ad^.a:=0;
  tree(ad,1);{Вызов процедуры создания дерева}
  i:=change(ad,1);{Переменной "i" присваивается верхнее ЗНАЧЕНИЕ узла дерева}
  write(i);
end.

```