Код программы

program ideone;

Type mass=array[1..150,1..150] of integer;  
Var i,j,n:integer;  
 a:mass;  
 t:boolean;  
 x,y:integer;  
  
Procedure printSquare(n:integer; a:mass);  
 Var i,j:integer;  
 Begin  
 for i:=1 to n do   
 begin  
 for j:=1 to n do write(a[i,j]:4,' ');  
 writeln();  
 end;  
 end;  
  
*{Магический квадрат n-го порядка при n=2m + 1 можно построить по правилу де Лялубера, суть которого заключается в следующем. Прежде всего число 1 помещается в среднюю клетку верхней строки.  
Последующие числа размещаются в их обычном порядке по направлению диагонали, идущей направо и вверх от данной клетки. При этом следует иметь в виду, что когда достигнута верхняя строка, следующее число нужно записать в нижнюю строку так, как если бы она была помещена над верхней строкой. При достижении крайнего правого столбца следующее число записывается в крайний левый столбец так, как если бы он был помещен непосредственно рядом с крайним правым столбцом.  
Когда требуемая для заполнения клетка уже занята или когда достигнута верхняя клетка крайнего правого столбца, необходимо спуститься по вертикали на строку вниз и затем продолжать заполнение по основному правилу}*

*//Создаем квадрат с нечетной стороной N*  
Procedure createOddSquare(n:integer; var a:mass);  
 Var i,j,k:integer;  
 p,l:integer;  
 Begin  
 for i:=1 to n do  
 for j:=1 to n do a[i,j]:=0;  
 j:=n div 2 +1; p:=sqr(n); i:=1; a[i,j]:=1;  
 for l:=2 to p do   
 begin  
 dec(i);  
 inc(j);  
 if (i=0) and (j<>n+1) then i:=n;  
 if (j=n+1) and (i<>0) then j:=1;  
 if ((i=0) and (j=n+1)) or (a[i,j]<>0) then  
 begin  
 inc(i,2);  
 dec(j);  
 end;  
 a[i,j]:=l;  
 end;  
 end;  
  
  
*{Магический квадрат n-го порядка при n=2(2m+1) можно построить по следующему правилу, которое разработал Р. Стрэчи в 1918 г. Разделим квадрат на четыре равных квадрата А, В, С, D.   
 A C   
 B D   
Построим в блоке А по правилу де Лялубера магический квадрат. Из чисел от 1 до u2, где u = n/2. Аналогичные магические квадраты построим в квадратах В, С, D соответственно из чисел: от u2+1 до 2u2, от 2u2 + 1 до Зu2 и от Зu2 + 1 до 4u2. Ясно, что получившийся в результате составной квадрат будет магическим по столбцам. В средней строке квадрата А возьмем m клеток от середины строки к левому краю, а в каждой из оставшихся строк возьмем m клеток, ближайших к левому краю квадрата A; числа в этих клетках поменяем местами с числами в соответствующих клетках квадрата В. Далее, возьмем числа в клетках каждого из m - 1 правых крайних столбцов квадрата С и поменяем их местами с соответствующими числами квадрата D. }*  
*//n=2(2m+1), где m - длина квадратов, на которые после разбивается матрица при построении*

*//Построение квадрата со стороной обычной четности*  
Procedure createSquareOfOrdinaryParity(n:integer; var a:mass);  
 Var u,i,j,k,m,z:integer;  
 b:mass;  
 Begin  
 u:= n div 2;  
 m:=(u-1) div 2;  
 *//Строим в левой верхней 1/4 части квадрат с нечетной стороной u*  
 createOddSquare(u,b);  
 k:=u\*u;  
 for i:=1 to n do  
 for j:=1 to n do  
 begin  
 if (i>=1) and (i<=u) and (j>=1) and (j<=u) then a[i,j]:=b[i,j];  
 if (i>=u+1) and (i<=n) and (j>=u+1) and (j<=n) then a[i,j]:=b[i-u,j-u]+k;  
 if (i>=1) and (i<=u) and (j>=u+1) and (j<=n) then a[i,j]:=b[i,j-u]+2\*k;  
 if (i>=u+1) and (i<=n) and (j>=1) and (j<=u) then a[i,j]:=b[i-u,j]+3\*k;  
 end;  
 for i:=1 to u do  
 if i=u div 2+1 then   
 begin  
 j:= u div 2+1;  
 for k:=1 to m do   
 begin  
 *//Меняем местами*  
 z:=a[i,j]; a[i,j]:=a[i+u,j]; a[i+u,j]:=z;  
 dec(j);  
 end;  
 end  
 else   
 begin  
 j:=1;  
 for k:=1 to m do   
 begin  
 *//Меняем местами*  
 z:=a[i,j]; a[i,j]:=a[i+u,j]; a[i+u,j]:=z;  
 inc(j);  
 end;  
 end;  
 j:=n;  
 for k:=1 to m-1 do   
 begin  
 for i:=1 to u do   
 begin  
 *//Меняем местами*  
 z:=a[i,j]; a[i,j]:=a[i+u,j]; a[i+u,j]:=z;  
 end;  
 dec(j);  
 end;  
 end;  
  
*{Магические квадраты порядка n=4m — квадраты двойной четности. Магический квадрат четвертого порядка можно построить путем выписывания чисел от I до 16 в их обычном порядке в четырех строках и последующей замены чисел, стоящих в диагональных клетках, дополнительными к ним числами, расположенными симметрично исходным числам относительно центра квадрата. Дополнительные числа вычисляются по формуле: n2-k+1, где k-число в ячейке. Диагональные клетки располагаются по четыре в шахматном порядке от центра. }*  
*//n=4m*

*//Построение квадрата со стороной двойной четности*  
Procedure createSquareDoubleParity(n:integer; var a:mass);  
 Var i,j,k:integer;  
 p,l:integer;  
 i1,j1,x,y:integer;  
 Begin  
 l:=1; p:=n\*n;  
 for i:=1 to n do  
 for j:=1 to n do   
 begin  
 a[i,j]:=l;  
 inc(l)  
 end;  
 i:=2;  
 while i<=n-2 do   
 begin  
 if i mod 4=0 then j:=4  
 else j:=2;  
 while j<=n-2 do   
 begin  
 for i1:=0 to 1 do  
 for j1:=0 to 1 do   
 begin  
 y:=i+i1; x:=j+j1;  
 a[y,x]:=p-a[y,x]+1;  
 end;  
 inc(j,4);  
 end;  
 inc(i,2);  
 end;  
 k:=4;  
 while k<=n-4 do   
 begin  
 a[1,k]:=p-a[1,k]+1; a[1,k+1]:=p-a[1,k+1]+1;  
 a[n,k]:=p-a[n,k]+1; a[n,k+1]:=p-a[n,k+1]+1;  
 a[k,1]:=p-a[k,1]+1; a[k+1,1]:=p-a[k+1,1]+1;  
 a[k,n]:=p-a[k,n]+1; a[k+1,n]:=p-a[k+1,n]+1;  
 inc(k,4);  
 end;  
 a[1,1]:=p-a[1,1]+1;  
 a[n,n]:=p-a[n,n]+1;  
 a[1,n]:=p-a[1,n]+1;  
 a[n,1]:=p-a[n,1]+1;  
 end;  
   
*//Проверяем суммы на строках, столбцах и диагоналях квадрата*  
Function test(n:integer; a:mass): boolean;  
 *//Массивы для записи сумм по строкам и столбцам*  
 Var s,z:array [1..150] of integer;  
 *//Сумма на диагоналях*  
 sd,zd:integer;  
 i,j,k:integer;  
 sum:integer;  
 t:boolean;  
 Begin  
 sum:=n\*(n\*n+1) div 2;  
 for k:=1 to n do   
 begin  
 s[k]:=0;  
 z[k]:=0  
 end;  
 sd:=0; zd:=0;  
 for i:=1 to n do  
 for j:=1 to n do   
 begin  
 s[i]:=s[i]+a[i,j];  
 z[j]:=z[j]+a[i,j]  
 end;  
 for k:=1 to n do   
 begin  
 sd:=sd+a[k,k];  
 zd:=zd+a[k,n-k+1];  
 end;  
 k:=1; t:=true;  
 while (k<=n) and (t) do   
 begin  
 *//Провереяем строки, столбцы, главную и побочную диагонали*   
 if (s[k]<>sum) or (z[k]<>sum) or (sd<>sum) or (zd<>sum) then t:=false;  
 inc(k);  
 end;  
 test:=t;  
 End;   
  
Begin  
  
Writeln('NxN - размерность квадрата. Введите N:');  
readln(n);  
  
if odd(n) then createOddSquare(n,a)  
else   
 if n mod 4=0 then createSquareDoubleParity(n,a)  
 else createSquareOfOrdinaryParity(n,a);  
  
*//Вывод результата*  
if(test(n,a)) then printSquare(n,a) else Writeln('Нельзя построить магический квадрат');  
  
end.

Тестирование программы

1. NxN - размерность квадрата. Введите N:

3  
 8 1 6   
 3 5 7   
 4 9 2

1. NxN - размерность квадрата. Введите N:

2

Нельзя построить магический квадрат

1. NxN - размерность квадрата. Введите N:

4  
 16 2 3 13   
 5 11 10 8   
 9 7 6 12   
 4 14 15 1

1. NxN - размерность квадрата. Введите N:

6  
 35 1 6 26 19 24   
 3 32 7 21 23 25   
 31 9 2 22 27 20   
 8 28 33 17 10 15   
 30 5 34 12 14 16   
 4 36 29 13 18 11

1. NxN - размерность квадрата. Введите N:

8  
 64 2 3 61 60 6 7 57   
 9 55 54 12 13 51 50 16   
 17 47 46 20 21 43 42 24   
 40 26 27 37 36 30 31 33   
 32 34 35 29 28 38 39 25   
 41 23 22 44 45 19 18 48   
 49 15 14 52 53 11 10 56   
 8 58 59 5 4 62 63 1