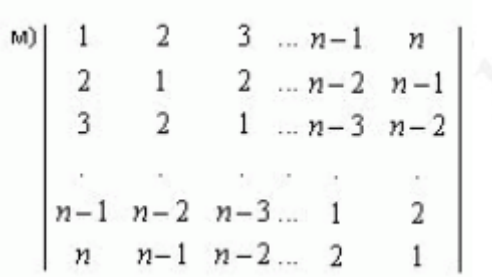
31 строка 695м

явно опечатка.

694 Получить квадратную матрицу порядка n:



Пример:

```
1 2 3 4 5 6 7 8
2 1 2 3 4 5 6 7
3 2 1 2 3 4 5 6
4 3 2 1 2 3 4 5
5 4 3 2 1 2 3 4
6 5 4 3 2 1 2 3
7 6 5 4 3 2 1 2
8 7 6 5 4 3 2 1
```pascal

Код

```pascal
unit Matrix;
interface
uses ...,ProcessMatrix;
type ...
var
  Form1: TForm1;
  TF:TextFile;

  OriginalMatrix:TMatrix;

  i,j:integer;
  n:integer;

  IBMessage:string;

implementation
{$R *.dfm}
begin
  n:=StrToInt(InputBox('степень матрицы','n:','2'));
  ProcessMatrix.CreateMatrix(OriginalMatrix,n,n);
  ProcessMatrix.ProcessingSpreading(OriginalMatrix,n);
  ProcessMatrix.OpenTextFile(TF);
  ProcessMatrix.WriteIntMatrixInTextFile(TF,OriginalMatrix);
```

```
37
38    CloseFile(TF);
39
40  end.
```

```
1  //unit ProcessMatrix;
2  procedure ProcessingSpreading(var Matrix:TMatrix;n:integer);
3  var i,j:integer;
4  begin
5  for i := 0 to length(Matrix)-1 do
6    for j := 0 to length(Matrix[1])-1 do
7      begin
8        if i=j then Matrix[i,j]:=1;
9        if i<j then Matrix[i,j]:=j-i+1;
10       if i>j then Matrix[i,j]:=-j+i+1
11     end;
12 end;
13
14 procedure WriteIntMatrixInTextFile(var TF:TextFile; Matrix:TMatrix);
15 var i,j: integer;
16 begin
17   Writeln(TF,' ');
18   Writeln(TF,'x:= '+IntToStr(length(Matrix[1]))+', y:=
   '+IntToStr(length(Matrix)));
19   for i := 0 to length(Matrix)-1 do
20   for j := 0 to length(Matrix[1])-1 do
21   begin
22     if (j <> length(Matrix[1])-1) then
23       Write(TF,Matrix[i,j]:1:0,' ')
24     else
25       Writeln(TF,Matrix[i,j]:1:0)
26   end;
27 end;
```

Вывод:

```
1  x:= 8, y:= 8
2  1 2 3 4 5 6 7 8
3  2 1 2 3 4 5 6 7
4  3 2 1 2 3 4 5 6
5  4 3 2 1 2 3 4 5
6  5 4 3 2 1 2 3 4
7  6 5 4 3 2 1 2 3
8  7 6 5 4 3 2 1 2
9  8 7 6 5 4 3 2 1
10
11 x:= 2, y:= 2
12 1 2
13 2 1
14
15 x:= 20, y:= 20
16 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
17 2 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
18 3 2 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
19 4 3 2 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
20 5 4 3 2 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
```

```
6  5  4  3  2  1  2  3  4  5  6  7  8  9  10  11  12  13  14  15
7  6  5  4  3  2  1  2  3  4  5  6  7  8  9  10  11  12  13  14
8  7  6  5  4  3  2  1  2  3  4  5  6  7  8  9  10  11  12  13
9  8  7  6  5  4  3  2  1  2  3  4  5  6  7  8  9  10  11  12
10  9  8  7  6  5  4  3  2  1  2  3  4  5  6  7  8  9  10  11
11  10  9  8  7  6  5  4  3  2  1  2  3  4  5  6  7  8  9  10
12  11  10  9  8  7  6  5  4  3  2  1  2  3  4  5  6  7  8  9
13  12  11  10  9  8  7  6  5  4  3  2  1  2  3  4  5  6  7  8
14  13  12  11  10  9  8  7  6  5  4  3  2  1  2  3  4  5  6  7
15  14  13  12  11  10  9  8  7  6  5  4  3  2  1  2  3  4  5  6
16  15  14  13  12  11  10  9  8  7  6  5  4  3  2  1  2  3  4  5
17  16  15  14  13  12  11  10  9  8  7  6  5  4  3  2  1  2  3  4
18  17  16  15  14  13  12  11  10  9  8  7  6  5  4  3  2  1  2  3
19  18  17  16  15  14  13  12  11  10  9  8  7  6  5  4  3  2  1  2
20  19  18  17  16  15  14  13  12  11  10  9  8  7  6  5  4  3  2  1
```