```
unit ProcessMatrix;

interface


uses
Windows, Messages, SysUtils, Classes, Graphics, Controls,
Forms, Dialogs, StdCtrls, ExtCtrls;

type
  TMatrix = array of array of Real;

procedure CreateMatrix(var Matrix:TMatrix; AxisY,AxisX: integer);
procedure GetCoord(var AxisY,AxisX: integer);
procedure InputMatrix(var Matrix:TMatrix);
  procedure RandomInputMatrix(var Matrix:TMatrix);
  procedure UserInputMatrix(var Matrix:TMatrix);

procedure OpenTextFile(var TF:TextFile);
  procedure ODOpenFile(var TF:TextFile);
  procedure PathOpenFile(var TF:TextFile);

procedure WriteMatrixInTextFile(var TF:TextFile; Matrix:TMatrix);
procedure WriteIntMatrixInTextFile(var TF:TextFile; Matrix:TMatrix);

procedure ProcessingFlip(var Matrix:TMatrix;y,x:integer);
procedure ProcessingSpreading(var Matrix:TMatrix;n:integer);

{ Dock
Input : User choice -> Random or Manual input
Output: User choice -> OpenDialog File or PopupPath File
}


implementation

procedure GetCoord(var AxisY,AxisX: integer);
const
  IBTitle = 'Создание матрицы';
  IBMessage1 = 'Ведите число строк: '; {строки - ось y}
  IBMessage2 = 'Ведите число столбцов: '; {столбцы - ось x}
  IBDefaultValue = '2';

begin
  AxisY:= StrToInt(InputBox(IBTitle,IBMessage1,IBDefaultValue));
  AxisX:= StrToInt(InputBox(IBTitle,IBMessage2,IBDefaultValue));
  if (AxisY<2) and (AxisX<2) then
  begin
    ShowMessage('Значения '+IntToStr(AxisY)+':y или '+IntToStr(AxisX)+':x
не коректны (y<2 или x<2)');
    AxisY:=2;AxisX:=2;
  end;


end;
```

```
55
56  procedure CreateMatrix(var Matrix:TMatrix; AxisY,AxisX: integer);
57  begin
58    SetLength(Matrix,AxisY,AxisX);
59  end;
60
61  procedure InputMatrix(var Matrix:TMatrix);
62  const
63    IBTitle = 'Создание матрицы';
64    IBMessage = 'Способ заполнения, Вручную-N, Автоматически-Y';
65    IBDefaultValue = 'Y';
66  var
67    flag:string;
68  begin
69    flag:= InputBox(IBTitle,IBMessage,IBDefaultValue);
70
71    if (flag = 'Y')or(flag = 'y')or(flag = ' ')or(flag = '') then
    RandomInputMatrix(Matrix)
72    else UserInputMatrix(Matrix);
73  end;
74
75  procedure RandomInputMatrix(var Matrix:TMatrix);
76  var i,j: integer;
77  begin
78    Randomize;
79      for i := 0 to length(Matrix)-1 do
80      for j := 0 to length(Matrix[1])-1 do
81      begin
82        Matrix[i,j]:= Random(10);
83        Matrix[i,j]:= Matrix[i,j] + Random;
84        Matrix[i,j]:= Round(matrix[i,j]*10000)/10000;
85      end;
86  end;
87
88  procedure UserInputMatrix(var Matrix:TMatrix);
89  const
90    IBTitle = 'Создание матрицы(2/2)';
91    IBDefaultValue = '1.0';
92  var
93    i,j: integer;
94    IBMessage: string;
95  begin
96    for i := 0 to length(Matrix)-1 do
97    for j := 0 to length(Matrix[1])-1 do
98    begin
99      IBMessage:='Введите matrix['+IntToStr(i)+']['+IntToStr(j)+'] элемент;';
100     Matrix[i,j]:=StrToFloat(InputBox(IBTitle,IBMessage,IBDefaultValue));
101   end;
102 end;
103
104 procedure OpenTextFile(var TF:TextFile);
105 const
106   IBTitle = 'Создание матрицы';
107   IBMessage = 'Способ выбора файла, Путь файла-N, Окно выбора-Y';
108   IBDefaultValue = 'Y';
109 var
110   s,Path,flag: string;
111   i:integer;
```

```
begin
  flag:= InputBox(IBTitle,IBMessage,IBDefaultValue);

  if (flag = 'Y')or(flag = 'y')or(flag = ' ')or(flag = '') then
ODOpenFile(TF)
  else PathOpenFile(TF);
end;

procedure WriteMatrixInTextFile(var TF:TextFile; Matrix:TMatrix);
var i,j: integer;
begin
  Writeln(TF,' ');
  Writeln(TF,'x:= '+IntToStr(length(Matrix[1]))+', y:=
'+IntToStr(length(Matrix)));
  for i := 0 to length(Matrix)-1 do
  for j := 0 to length(Matrix[1])-1 do
  begin
    if (j <> length(Matrix[1])-1) then
      Write(TF,Matrix[i,j]:8:4,' ')
    else
      Writeln(TF,Matrix[i,j]:8:4)
  end;
end;

procedure WriteIntMatrixInTextFile(var TF:TextFile; Matrix:TMatrix);
var i,j: integer;
begin
  Writeln(TF,' ');
  Writeln(TF,'x:= '+IntToStr(length(Matrix[1]))+', y:=
'+IntToStr(length(Matrix)));
  for i := 0 to length(Matrix)-1 do
  for j := 0 to length(Matrix[1])-1 do
  begin
    if (j <> length(Matrix[1])-1) then
      Write(TF,Matrix[i,j]:1:0,' ')
    else
      Writeln(TF,Matrix[i,j]:1:0)
  end;
end;

procedure ODOpenFile(var TF:TextFile);
var
  Path:string;
  openDialog : TOpenDialog;
begin
  openDialog:=TOpenDialog.Create(nil);
  if openDialog.Execute then
  if FileExists(openDialog.FileName) then
  Path:=openDialog.FileName
  else raise Exception.Create('File does not exist.');

  openDialog.Free;
  AssignFile(TF,Path);
  if FileExists(Path) then Append(TF) else Rewrite(TF);


end;

```

```
procedure PathOpenFile(var TF:TextFile);
var Path:string;
begin
  Path:=InputBox('Введите путь','','A:\Desktop\');
  AssignFile(TF,Path);
  if FileExists(Path) then Append(TF) else Rewrite(TF);
end;


procedure ProcessingFlip(var Matrix:TMatrix;y,x:integer);
var centerX,centerY,i,j:integer;
    TempVar:real;
begin
  centerX:=((x-1) div 2);
  centerY:=((y-1) div 2);
  for i:=0 to centerY do
    for j:=0 to centerX do
      begin
        TempVar:=Matrix[i,j];
        Matrix[i,j]    :=Matrix[centerY+i+1,centerX+j+1];
        Matrix[centerY+i+1,centerX+j+1]:=TempVar;
      end;

  for i:=0 to centerY do
    for j:=0 to centerX do
      begin
        TempVar:=Matrix[centerY+i+1,j];
        Matrix[centerY+i+1,j]    :=Matrix[i,centerX+j+1];
        Matrix[i,centerX+j+1]:=TempVar;
      end;
end;

procedure ProcessingSpreading(var Matrix:TMatrix;n:integer);
var i,j:integer;
begin
for i := 0 to length(Matrix)-1 do
  for j := 0 to length(Matrix[1])-1 do
    begin
      if i=j then Matrix[i,j]:=1;
      if i<j then Matrix[i,j]:=j-i+1;
      if i>j then Matrix[i,j]:=-j+i+1
    end;
end;

end.

```