



BRANCHE	SECTION	ÉPREUVE ÉCRITE
INFORMATIQUE partie pratique	B	<i>Durée de l'épreuve : 80 minutes</i> <i>Date de l'épreuve :</i>

```
unit UMain;
interface
uses Windows, SysUtils, Classes, Graphics, Controls, Forms, StdCtrls, Grids, ExtCtrls;
//-----
type
TfrmMain = class(TForm)
  imgDessin: TImage;          sgGrille: TStringGrid;
  lbValeurs: TListBox;        btnTransferer: TButton;
  btnEncoder: TButton;       btnInverser: TButton;
  btnNouveau: TButton;     btnAjouter: TButton;
  procedure FormCreate(Sender: TObject);
  procedure btnNouveauClick(Sender: TObject);
  procedure imgDessinMouseDown(Sender: TObject; Button: TMouseButton;
    Shift: TShiftState; X, Y: Integer);
  procedure btnAjouterClick(Sender: TObject);
  procedure btnInverserClick(Sender: TObject);
  procedure btnTransfererClick(Sender: TObject);
  procedure btnEncoderClick(Sender: TObject);
end;

var
  frmMain: TfrmMain;

implementation
{$R *.DFM}
//-----
procedure TfrmMain.FormCreate(Sender: TObject);
begin
  randomize;
  imgDessin.Canvas.Pixels[-1,-1] := clWhite;
end;
//-----
procedure Transfert(pImage:TImage;var pGrille:TStringGrid);
var LI, CO : integer;
begin
  pGrille.RowCount := pImage.Height;
  pGrille.ColCount := pImage.Width;
  for LI := 0 to pImage.Height-1 do
    for CO := 0 to pImage.Width-1 do
      if pImage.Canvas.Pixels[CO,LI] = clBlack
      then pGrille.Cells[CO,LI] := 'n'
      else pGrille.Cells[CO,LI] := 'b'
    end;
  end;
//-----
procedure Encodage(pGrille:TStringGrid;var pListe:TListBox);
var LI, CO, HAUTEUR, LARGEUR, I, COUNT : integer;
  PRECEDENT, VALEUR : string;
begin
  LARGEUR := pGrille.ColCount;
  HAUTEUR := pGrille.RowCount;
  pListe.Items.Clear;
  pListe.Items.Append(inttostr(LARGEUR));
  pListe.Items.Append(inttostr(HAUTEUR));
  COUNT := 0;
  PRECEDENT := 'n';
  for I := 0 to LARGEUR * HAUTEUR - 1 do begin
    LI := I div LARGEUR;
    CO := I mod LARGEUR;
    VALEUR := pGrille.Cells[CO,LI];
    if (VALEUR <> PRECEDENT) or (I = LARGEUR * HAUTEUR)
    then begin
      pListe.Items.Append(inttostr(COUNT));
      COUNT := 1;
      PRECEDENT := VALEUR;
    end
    else COUNT := COUNT + 1;
  end;
end;
//-----
```

```
//-----
procedure TfrmMain.btnNouveauClick(Sender: TObject); // 3 p.
var CO, LI : integer;
begin
  imgDessin.Canvas.Brush.Color := clWhite;
  imgDessin.Canvas.Pen.Color := clWhite;
  imgDessin.Canvas.Rectangle(0,0,imgDessin.Width,imgDessin.Height);
  lbValeurs.Items.Clear;
  sgGrille.RowCount := 5;
  sgGrille.ColCount := 5;
  for CO := 0 to 4 do
    for LI := 0 to 4 do
      sgGrille.Cells[CO,LI] := '';
  end;
//-----
procedure TfrmMain.imgDessinMouseDown(Sender: TObject; Button: TMouseButton; // 3 p.
  Shift: TShiftState; X, Y: Integer);
begin
  imgDessin.Canvas.Pen.Color := clBlack;
  imgDessin.Canvas.Brush.Color := clBlack;
  if Button = mbLeft then imgDessin.Canvas.Ellipse(x-5,y-5,x+6,y+6);
  if Button = mbRight then imgDessin.Canvas.Rectangle(x-7,y-3,x+8,y+4);
end;
//-----
procedure TfrmMain.btnAjouterClick(Sender: TObject); // 4 p.
var I, X, Y : integer;
begin
  imgDessin.Canvas.Pen.Color := clBlack;
  imgDessin.Canvas.Brush.Color := clBlack;
  for I := 1 to 6 do begin // on accepte ici qu'on prenne le maximum
    X := random(imgDessin.Width-15) + 7; // commun aux deux formes
    Y := random(imgDessin.Height-11) + 5;
    if random(2) = 1
    then imgDessin.Canvas.Ellipse(X-5,Y-5,X+6,Y+6)
    else imgDessin.Canvas.Rectangle(X-7,Y-3,X+8,Y+4);
  end;
end;
//-----
procedure TfrmMain.btnInverserClick(Sender: TObject); // 3 p.
var LI, CO : integer;
begin
  for LI := 0 to imgDessin.Height-1 do
    for CO := 0 to imgDessin.Width-1 do
      if imgDessin.Canvas.Pixels[CO,LI] = clWhite
      then imgDessin.Canvas.Pixels[CO,LI] := clBlack
      else imgDessin.Canvas.Pixels[CO,LI] := clWhite;
  end;
//-----
procedure TfrmMain.btnTransfererClick(Sender: TObject); // 1 p.
begin
  Transfert(imgDessin,sgGrille);
end;
//-----
procedure TfrmMain.btnEncoderClick(Sender: TObject); // 1 p.
begin
  Encodage(sgGrille,lbValeurs);
end;
//-----
end.
```