



ÉPREUVE ÉCRITE – partie pratique	Branche : INFORMATIQUE
Section(s) : B	N° d'ordre du candidat :
Date de l'épreuve :	Durée de l'épreuve : 80 minutes

```
1  unit UMain;
2
3  interface
4
5  uses
6      Forms, SysUtils, Graphics, Controls, StdCtrls, Grids, Classes, ExtCtrls;
7  //-----
8  Type                                // interface et nomenclature    4 p.
9      TfrmMain = class(TForm)
10         sgAmpoules: TStringGrid;
11         btnAjouter: TButton;
12         btnInverser: TButton;
13         btnVider: TButton;
14         imgDessin: TImage;
15         procedure btnAjouterClick(Sender: TObject);
16         procedure FormCreate(Sender: TObject);
17         procedure btnInverserClick(Sender: TObject);
18         procedure btnViderClick(Sender: TObject);
19         procedure imgDessinMouseDown(Sender: TObject; Button: TMouseButton;
20             Shift: TShiftState; X, Y: Integer);
21     end;
22 //-----
23 var
24     frmMain: TfrmMain;
25
26 implementation
27
28 {$R *.DFM}
29 //-----
30 procedure TfrmMain.FormCreate(Sender: TObject);                                // 2 p.
31 begin
32     randomize;
33     sgAmpoules.Cells[0,0] := 'x';
34     sgAmpoules.Cells[1,0] := 'y';
35     sgAmpoules.Cells[2,0] := 'état';
36     imgDessin.Canvas.Brush.Style := bsSolid;
37     imgDessin.Canvas.Brush.Color := clWhite;
38     imgDessin.Canvas.Rectangle(0,0,imgDessin.Width,imgDessin.Height);
39 end;
40 //-----
41
```



```

42 //-----
43 procedure TfrmMain.btnAjouterClick(Sender: TObject); // 7 p.
44 var i, x, y, state, r : integer;
45 begin
46   for I := 1 to 5 do begin
47     x := random(imgDessin.Width-17)+8;
48     y := random(imgDessin.Height-24);
49     state := random(2);
50     r := sgAmpoules.RowCount-1;
51     if sgAmpoules.Cells[0,r] <> ''
52     then r := r + 1;
53     sgAmpoules.RowCount := r+1;
54     sgAmpoules.Cells[0,r] := inttostr(x);
55     sgAmpoules.Cells[1,r] := inttostr(y);
56     sgAmpoules.Cells[2,r] := inttostr(state);
57     imgDessin.Canvas.Brush.Style := bsSolid;
58     imgDessin.Canvas.Brush.Color := clBlack;
59     imgDessin.Canvas.Pen.Color := clBlack;
60     imgDessin.Canvas.Rectangle(x-3, y, x+4,y+8);
61     if (state=1)
62     then imgDessin.Canvas.Brush.Color := clYellow
63     else imgDessin.Canvas.Brush.Color := clWhite;
64     imgDessin.Canvas.Ellipse(x-8,y+7,x+9,y+25);
65     imgDessin.Canvas.Ellipse(x-8,y+7,x+9,y+25);
66   end;
67 end;
68 //-----
69 procedure TfrmMain.btnInverserClick(Sender: TObject); // 7 p.
70
71 var i, x, y, state : integer;
72 begin
73   if sgAmpoules.Cells[0,1] <> '' then begin
74     imgDessin.Canvas.Brush.Style := bsSolid;
75     imgDessin.Canvas.Brush.Color := clWhite;
76     imgDessin.Canvas.Rectangle(0,0,imgDessin.Width,imgDessin.Height);
77
78     for I := 1 to sgAmpoules.RowCount - 1 do begin
79       sgAmpoules.Cells[2,i] := inttostr(1 - strtoint(sgAmpoules.Cells[2,i]));
80
81       x := strtoint(sgAmpoules.Cells[0,i]);
82       y := strtoint(sgAmpoules.Cells[1,i]);
83       state := strtoint(sgAmpoules.Cells[2,i]);
84       imgDessin.Canvas.Brush.Style := bsSolid;
85       imgDessin.Canvas.Brush.Color := clBlack;
86       imgDessin.Canvas.Pen.Color := clBlack;
87       imgDessin.Canvas.Rectangle(x-3, y, x+4,y+8);
88       if (state=1)
89       then imgDessin.Canvas.Brush.Color := clYellow
90       else imgDessin.Canvas.Brush.Color := clWhite;
91       imgDessin.Canvas.Ellipse(x-8,y+7,x+9,y+25);
92     end;
93   end;
94 end;
95 //-----
96 procedure TfrmMain.btnViderClick(Sender: TObject); // 2 p.
97
98 begin
99   sgAmpoules.RowCount := 2;
100   sgAmpoules.Cells[0,1] := '';
101   sgAmpoules.Cells[1,1] := '';
102   sgAmpoules.Cells[2,1] := '';
103   imgDessin.Canvas.Brush.Style := bsSolid;
104   imgDessin.Canvas.Brush.Color := clWhite;
105   imgDessin.Canvas.Rectangle(0,0,imgDessin.Width,imgDessin.Height);
106 end;
107 //-----
108

```



```

109 //-----
110 procedure TfrmMain.imgDessinMouseDown(Sender: TObject;                               // 8 p.
111
112     Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
113     var lx, ly, state, i : integer;
114 begin
115     if (sgAmpoules.Cells[0,1] <> '') and (Button = mbLeft) then begin
116         for I := 1 to sgAmpoules.RowCount - 1 do begin
117             lx := strtoint(sgAmpoules.Cells[0,i]);
118             ly := strtoint(sgAmpoules.Cells[1,i]);
119             state := strtoint(sgAmpoules.Cells[2,i]);
120             if (X >=lx-3) and (X <=lx+3) and (Y >=ly) and (Y <=ly+7) then begin
121                 state := 1-state;
122                 imgDessin.Canvas.Brush.Style := bsSolid;
123                 imgDessin.Canvas.Brush.Color := clBlack;
124                 imgDessin.Canvas.Pen.Color := clBlack;
125                 imgDessin.Canvas.Rectangle(lx-3, ly, lx+4,ly+8);
126                 if (state=1)
127                 then imgDessin.Canvas.Brush.Color := clYellow
128                 else imgDessin.Canvas.Brush.Color := clWhite;
129                 imgDessin.Canvas.Ellipse(lx-8,ly+7,lx+9,ly+25);
130                 sgAmpoules.Cells[2,i] := inttostr(state);
131             end;
132         end;
133     end;
134 end;
135 //-----
136 end.

```