

## Épreuve pratique – CORRIGÉ MODÈLE

**Examen de fin d'études secondaires 2010**

**Section:**

**Branche: Informatique**

**Numéro d'ordre du candidat**

---

```
1 unit Nombres_UMain;
2 interface
3 uses
4   Windows, Messages, SysUtils, Classes, Controls, Forms, StdCtrls, Grids;
5 //-----
6 type
7   TfrmMain = class(TForm)
8     lblTaille: TLabel;
9     lblChaine: TLabel;
10    edtTaille: TEdit;
11    btnGenerer: TButton;
12    btnDecouper: TButton;
13    btnSupprimer: TButton;
14    btnTransferer: TButton;
15    btnQuitter: TButton;
16    lbValeurs: TListBox;
17    sgMatrice: TStringGrid;
18    procedure btnGenererClick(Sender: TObject);
19    procedure btnQuitterClick(Sender: TObject);
20    procedure btnDecouperClick(Sender: TObject);
21    procedure btnTransfererClick(Sender: TObject);
22    procedure btnSupprimerClick(Sender: TObject);
23    procedure FormCreate(Sender: TObject);
24  private
25    { Private declarations }
26  public
27    { Public declarations }
28  end;
29
30 var
31   frmMain: TfrmMain;
32
33 implementation
34 {$R *.DFM}
35
36 //-----
37 function ALEA(T:integer):string;
38 begin
39   Result := '';
40   while T > 0 do begin
41     Result := Result + inttostr(random(9) + 1) + inttostr(random(9) + 1);
42     T := T - 1;
43   end;
44 end;
45
46 //-----
47 procedure TfrmMain.FormCreate(Sender: TObject);
48 begin
49   randomize;
50 end;
```

## Épreuve pratique – CORRIGÉ MODÈLE

```
47 //-----
48 procedure TfrmMain.btnGenererClick(Sender: TObject);           // 2 p.
49 begin
50   lblChaine.Caption := ALEA(strtoint(edtTaille.Text));
51 end;
52
53 //-----
54 procedure TfrmMain.btnDecouperClick(Sender: TObject);         // 5 p.
55 var S : string;
56 begin
57   lbValeurs.Clear;
58   S := lblChaine.Caption;
59   while S <> '' do begin
60     lbValeurs.Items.Append(Copy(S,1,2));
61     Delete(S,1,2);
62   end;
63 end;
64
65 //-----
66 procedure TfrmMain.btnSupprimerClick(Sender: TObject);       // 6 p.
67 var I, J : integer;
68 begin
69   I := 0;
70   while I < lbValeurs.Items.Count do begin
71     for J := lbValeurs.Items.Count-1 downto I+1 do
72       if lbValeurs.Items[I] = lbValeurs.Items[J]
73         then lbValeurs.Items.Delete(J);
74     I := I + 1;
75   end;
76 end;
77
78 //-----
79 procedure TfrmMain.btnTransfererClick(Sender: TObject);     // 7 p.
80 var I, N : integer;
81 begin
82   N := round(sqrt(lbValeurs.Items.Count)+0.49999999);
83   sgMatrice.Colcount := N;
84   sgMatrice.Rowcount := N;
85   for I := 0 to sqr(N) - 1 do
86     if I < lbValeurs.Items.Count
87       then sgMatrice.Cells[I mod N, I div N] := lbValeurs.Items[I]
88       else sgMatrice.Cells[I mod N, I div N] := '';
89 end;
90
91 //-----
92 procedure TfrmMain.btnQuitterClick(Sender: TObject);        // 1 p.
93 begin
94   Close;
95 end;
96
97 //-----
98 end.
```