Question 1

(1) a) On déclare un nouveau type :

```
poly = record
  coeff : array[0..100] of extended;
  degre : integer
end;
```

Bien sûr, on est alors limité à des polynômes de degré ≤ 100.

b) On crée un composant `lbP` de type `TListBox`, dont les éléments seront les coefficients du polynôme. Le degré du polynôme est alors intrinsèquement accessible via `lbP.Items.Count-1`.

Question 2

Voici l'algorithme corrigé :

```
MIN :=60;
for I:=0 to lbListe.Items.Count-1 do
  if StrToInt(lbListe.Items[I])<MIN then
    MIN:=StrToInt(lbListe.Items[I]);
lblResultat.Caption:='Note minimale : '+IntToStr(MIN);
```

Question 3

(1) $\text{exem}(5,-3) = 2$

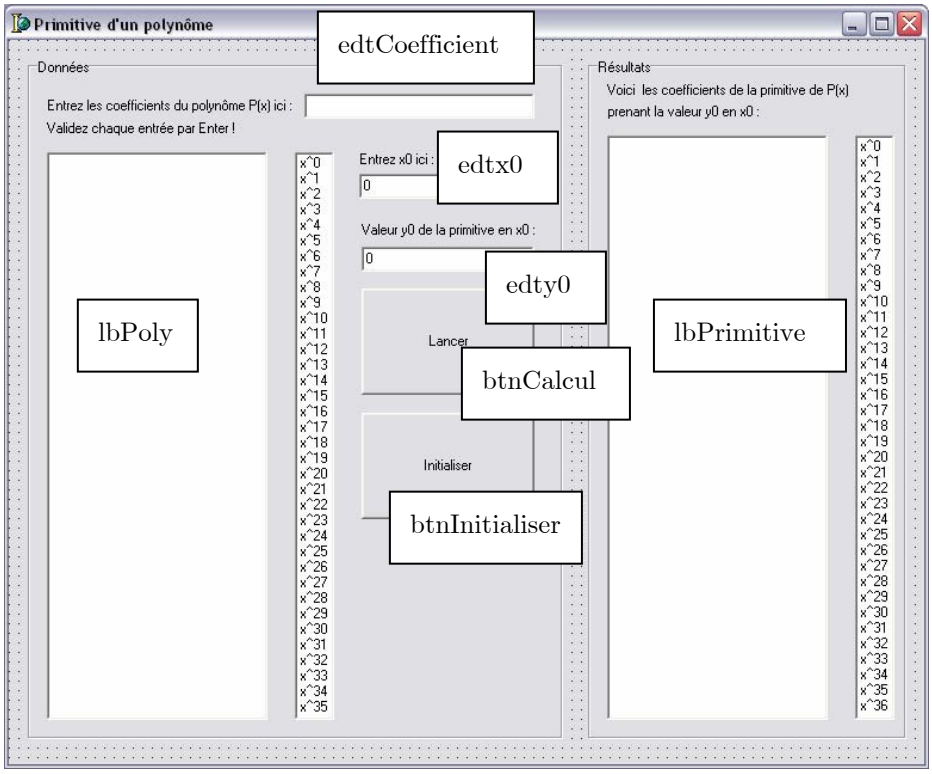
(2) $\begin{aligned}\text{exem}(7,5) &= 35 + \text{exem}(6,-2)\\ &= 35 + 2 = 37\end{aligned}$

(3) $\begin{aligned}\text{exem}(1,10) &= 10 + \text{exem}(0,9)\\ &= 10 + 2\\ &= 12\end{aligned}$

Question 4

```
implementation
{$R *.dfm}

procedure Primitive(lbA,lbPrim:TListBox);
{La procédure donne la primitive à terme constant 0.}
var dA,i:integer;
    x:real;
begin
  lbPrim.Clear;
  lbPrim.Items[0]:='0';
  dA:=lbA.Items.Count-1;
  for i:=0 to dA do begin
                      x:=StrToFloat(lbA.Items[i]);
                      lbPrim.Items[i+1]:=FloatToStr(x/(i+1));
                    end;
end;
```




```
function Horner(lbA:TListBox;x:real):real;
var dA,i:integer;
    res:real;
begin
  dA:=lbA.Items.Count-1;
  res:=0;
  for i:=dA downto 0 do res:=res*x+StrToFloat(lbA.Items[i]);
  Horner:=res
end;

procedure TfrmIntegration.btnInitialiserClick(Sender: TObject);
begin
  lbPoly.Clear;
  lbPrimitive.Clear;
  edtCoefficient.Clear;
  edtx0.Text:='0';
  edty0.Text:='0';
  edtCoefficient.SetFocus
end;

procedure TfrmIntegration.edtCoefficientKeyDown(Sender: TObject;
  var Key: Word; Shift: TShiftState);
begin
  if Key=VK_RETURN then begin
                          lbPoly.Items.Add(edtCoefficient.Text);
                          edtCoefficient.Text:='';
                          edtCoefficient.SetFocus
                        end
end;

procedure TfrmIntegration.btnCalculClick(Sender: TObject);
var k,x0,y0:real;
begin
  Primitive(lbPoly,lbPrimitive);
  x0:=StrToFloat(edtx0.text);
  y0:=StrToFloat(edty0.text);
  k:=y0-Horner(lbPrimitive,x0);
  lbPrimitive.Items[0]:=FloatToStr(k)
end;
end.
```