

## Question 1

- (1) La liste doit être triée !
- (2) Voir document « Algorithmes obligatoires » (version corrigée) :

```

function dichotomie(liste:TListbox;cle:string):integer;
var milieu,g,d:integer;
begin
  g:=0;
  d:=liste.Items.Count-1;
  milieu :=(g+d) div 2 ;
  while (cle<>liste.Items[milieu]) and (g<=d) do begin
    if cle<liste.Items[milieu] then d:=milieu-1
      else g:=milieu+1;
    milieu :=(g+d)div 2;
  end;
  if cle=liste.Items[milieu] then dichotomie:=milieu
  else dichotomie:=-1
end;

```

- (3) L'évolution des variables **g**, **milieu** et **d** est donnée dans le tableau suivant :

g	milieu	d
0	12	25
13	12	25
13	19	25
20	19	25
20	22	25
20	22	21
20	20	21

## Question 2

```

1 function maximum(liste:TListbox):integer;
2 var indice,maxi:integer;
3 begin
4   maxi:=StrToInt(liste.Items[0]);
5   for indice:=1 to liste.Items.Count-1 do
6     if StrToInt(liste.Items[indice])>maxi then
7       maxi:=StrToInt(liste.Items[indice]);
8   maximum:=maxi
9 end;

```

```

1 function minimum(liste:TListbox):integer;
2 var indice,mini:integer;
3 begin
4   mini:=StrToInt(liste.Items[0]);
5   for indice:=1 to liste.Items.Count-1 do
6     if StrToInt(liste.Items[indice])<mini then
7       mini:=StrToInt(liste.Items[indice]);
8   minimum:=mini
9 end;

```

Dans l'algorithme sur le minimum, il faudra bien entendu faire les changements adéquats à propos du type réel.

### Question 3

(1) Interface graphique :

(2)

```
procedure inverse(var x, y: real);  
var denom: real;  
begin  
    denom:=x*x+y*y;  
    x:=x/denom;  
    y:=-y/denom;  
end;
```

(3)

```
procedure harmonic(x1, y1, x2, y2: real; var x, y: real);  
begin  
    inverse(x1, y1);  
    inverse(x2, y2);  
    x:=(x1+x2)/2;  
    y:=(y1+y2)/2;  
    inverse(x, y);  
end;
```

(4)

```
procedure suite(Lreal, Limaginary: TListBox; n: integer);  
var  
    i, j: integer;  
    x1, y1, x2, y2, x, y: real;  
begin  
    randomize;  
    Lreal.Clear;  
    Limaginary.Clear;  
    // initialisation de la suite  
    Lreal.Items.Append(floattostr(0));  
    Lreal.Items.Append(floattostr(-0.9));  
    Lreal.Items.Append(floattostr(0.9));  
    Lreal.Items.Append(floattostr(0));  
    Limaginary.Items.Append(floattostr(1));  
    Limaginary.Items.Append(floattostr(-0.5));  
    Limaginary.Items.Append(floattostr(-0.5));  
    Limaginary.Items.Append(floattostr(1));  
    // moyennes harmoniques  
    for j:=4 to n do  
        begin  
            i:=Lreal.Count-1; // i est le dernier element de la liste  
            x1:=strtofloat(Lreal.Items[i]);  
            y1:=strtofloat(Limaginary.Items[i]);  
            i:=random(3); // l'un des 3 premiers elements est choisi au hasard  
            x2:=strtofloat(Lreal.Items[i]);  
            y2:=strtofloat(Limaginary.Items[i]);  
            harmonic(x1, y1, x2, y2, x, y);  
            Lreal.Items.Append(floattostr(x));  
            Limaginary.Items.Append(floattostr(y));  
        end;  
    end;  
  
procedure TForm1.btnStartClick(Sender: TObject);  
var n: integer;  
begin  
    n:=strtoint(Edit1.Text);  
    suite(lbReal, lbImaginary, n);  
end;  
  
end.
```

