

Correction de la partie théorique :

Question 1

- (1) On recherche le plus petit élément et on l'échange avec le premier élément de la liste, ensuite on recherche le plus petit élément parmi les éléments restants (c.-à-d. à partir du deuxième élément de liste) et on l'échange avec le second élément de la liste etc., jusqu'à ce que la liste soit entièrement triée. Donc, à chaque fois qu'on recherche l'élément le plus petit, on n'a plus besoin de parcourir les éléments déjà triés.
- (2) Voir cours.
- (3) ('C', 'D', 'B', 'A')
('A', 'D', 'B', 'C')
('A', 'B', 'D', 'C')
('A', 'B', 'C', 'D')

Question 2

- (1) Voir cours.
- (2)

```
function puissance(var p:poly;expo:integer):poly;
var q:poly;
begin
  if expo=0 then begin
    result.d:=0;
    result.c[0]:=1
  end
  else if expo mod 2=0 then
    begin
      q:=produit(p,p);
      result:=puissance(q,expo div 2)
    end
  else begin
    q:=puissance(p,expo-1);
    result:=produit(p,q);
  end
end;
```

Question 3

- (1) a) mystere('musique', 'mathematiques')
= mystere('usique', 'mathematiques')
= mystere('sique', 'mathematiques')
= mystere('ique', 'mathematiques')
= mystere('que', 'mathematiques')
= mystere('ue', 'mathematiques')
= mystere('e', 'mathematiques')
= mystere('', 'mathematiques')
= true
b) mystere('musicien', 'instrument')
= mystere('usicien', 'instrument')
= mystere('sicien', 'instrument')
= mystere('icien', 'instrument')
= mystere('cien', 'instrument')
= false
- (2) x= 'cba' ou x= 'abbac' ou x= 'aabbcccbaba' par exemple.
- (3) mystere(a,b) retourne true si l'ensemble des lettres du string a est inclus dans l'ensemble des lettres du string b, et false sinon.

Correction de la partie pratique :

```
unit MusicU;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
  Forms,
  Dialogs, Grids, StdCtrls;

type
  TForm1 = class(TForm)
    Label1: TLabel;
    edtArtist: TEdit;
    btnAdd: TButton;
    btnDelete: TButton;
    sgM: TStringGrid;
    btnCD: TButton;
    edtRow: TEdit;
    Label2: TLabel;
    edtSong: TEdit;
    lblLength: TLabel;
    Label3: TLabel;
    lblNumber: TLabel;
    procedure btnAddClick(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure btnDeleteClick(Sender: TObject);
    procedure btnCDClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

implementation

{$R *.dfm}

//Question 2

procedure TForm1.FormCreate(Sender: TObject);
begin
  randomize;
  sgM.Cells[0,0]:='Number';
  sgM.Cells[1,0]:='Artist';
  sgM.Cells[2,0]:='Song';
  sgM.Cells[3,0]:='Length';
  sgM.Cells[4,0]:='CD';
end;

//Question 3

procedure TForm1.btnAddClick(Sender: TObject);
var n_songs:integer;
begin
  n_songs:=strtoint(lblNumber.Caption)+1;
  lblNumber.Caption:=inttostr(n_songs);
```

```

sgM.RowCount:=sgM.RowCount+1;
sgM.Cells[0,n_songs]:=inttostr(n_songs);
sgM.Cells[1,n_songs]:=edtArtist.Text;
sgM.Cells[2,n_songs]:=edtSong.Text;
sgM.Cells[3,n_songs]:=
    inttostr(random(20))+ 'm'+inttostr(random(60))+ 's'
end;

```

//Question 4

```

procedure TForm1.btnDeleteClick(Sender: TObject);
var i,j,n_delete,n_songs:integer;
begin
    n_songs:=strtoint(lblNumber.Caption);
    n_delete:=strtoint(edtRow.Text);
    if (n_delete>0) and (n_delete<=n_songs) then
        begin
            for i:=n_delete+1 to n_songs do
                for j:=1 to 4 do sgM.Cells[j,i-1]:=sgM.Cells[j,i];
                for j:=0 to 4 do sgM.Cells[j,n_songs]:='';
                lblNumber.Caption:=inttostr(n_songs-1);
                sgM.RowCount:=sgM.RowCount-1;
            end
            else showmessage('Numéro non valide !')
        end;
end;

```

//Question 5

```

function toSeconds(s:string):integer;
var m:integer;
begin
    m:=pos('m',s);
    result:=strtoint(copy(s,1,m-1))*60+
        strtoint(copy(s,m+1,length(s)-m-1));
end;

```

```

function min(sgM:TStringGrid):integer;
var i:integer;
begin
    result:=4441;
    i:=1;
    while sgM.Cells[1,i]<>' ' do
        begin
            if (sgM.Cells[4,i]=' ') and
                (toSeconds(sgM.Cells[3,i])<result)
            then result:=toSeconds(sgM.Cells[3,i]);
            i:=i+1
        end;
    end;
end;

```

```

procedure TForm1.btnCDCClick(Sender: TObject);
var i,n_songs,long_tot,item,long_item:integer;
begin

```

```

    n_songs:=strtoint(lblNumber.Caption);
    for i:=1 to n_songs do sgM.Cells[4,i]:='';
    lblLength.Caption:='Total length : ';
    long_tot:=0;
    while min(sgM)+long_tot<=4440 do
//Lorsqu'il n'y a plus d'éléments disponibles, min(sgM)=4441, donc on sort
de la boucle

```

```
begin
  item:=random(n_songs)+1;
  long_item:=toSeconds(sgM.Cells[3,item]);
  if (sgM.Cells[4,item]='') and (long_tot+long_item<=4440) then
    begin
      sgM.Cells[4,item]='X';
      long_tot:=long_tot+long_item
    end
  end;
  lblLength.Caption:='Total length : '
  +inttostr(long_tot div 60)+'m'+inttostr(long_tot mod 60)+'s'
end;

end.
```