

## Projet: Exercice 3.5 fonction FromDecimal version 1.0

Converti de décimal dans les base 2, 8, 16.

Le nombre à convertir est dans EditDecimal.

Paramètre base : La base du résultat.

Paramètre de sortie: Le résultat sous forme de chaîne de caractères.

Initialisations du reste et du résultat

tant que le nombre > 0

Reste de la division par la base désirée

Le reste est-il plus grand que 9 ?

oui

non

Conversion en lettre

A = 10, F = 15

Concaténation dans la chaîne résultat

Concaténation dans la chaîne résultat

Division de nombre par la base désirée



```

{*****}
Projet      : Exercie 3.5
Auteur      : J Court
Description  : Changement de base Decimal,octal, binaire hexadécimal
Version     : 1.0 9/9/2002
{*****}

unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, ExtCtrls;
type
  TForm1 = class(TForm)
    EditOct: TEdit;
    Button1: TButton;
    EditBin: TEdit;
    EditDecimal: TEdit;
    Binaire: TLabel;
    Octal: TLabel;
    Label1: TLabel;
    EditHexa: TEdit;
    Label2: TLabel;
    procedure Button1Click(Sender: TObject);
    procedure EditBinKeyUp(Sender: TObject; var Key: Word;
      Shift: TShiftState);
    procedure EditOctKeyUp(Sender: TObject; var Key: Word;
      Shift: TShiftState);
    procedure EditHexaKeyUp(Sender: TObject; var Key: Word;
      Shift: TShiftState);
    procedure EditDecimalKeyUp(Sender: TObject; var Key: Word;
      Shift: TShiftState);
    procedure EditBinKeyPress(Sender: TObject; var Key: Char);
    procedure EditOctKeyPress(Sender: TObject; var Key: Char);
    procedure EditDecimalKeyPress(Sender: TObject; var Key: Char);
    procedure EditHexaKeyPress(Sender: TObject; var Key: Char);
  private
    function FromDecimal(base : integer):string;
    procedure ToDecimal(base : integer; StrIn : string);
    procedure EffaceTout;
  { Private declarations }
  public
    { Public declarations }
  end;
var
  Form1: TForm1;
implementation
  {$R *.DFM}

  ===== constantes globales =====

const BACK_SPACE = #8;           // valeur du code ASCII
const NUL         = #0;           // valeur du code ASCII
const BASE_2      = 2;
const BASE_8      = 8;
const BASE_16     = 16;

//===== Sortie du programme =====

procedure TForm1.Button1Click(Sender: TObject);
begin
  application.terminate;
end;
{===== Tform1.EffaceTout =====}
Description : Efface les 4 edits decimal.octal,hexadécimal et binaire
Paramètre   : Aucun
-----}

procedure TForm1.EffaceTout;
begin
  EditBin.clear;
  EditOct.clear;
  EditHexa.clear;
  EditDecimal.clear;
end;

```



```
{===== Tform1.FromDecimal =====}
Description : Converti de décimal en une base 2, 8 16.
               Le nombre à convertir est pris dans EditDecimal
Paramètre   : Base  => La base de destination
retour      : result => sous forme de string (chaîne de caractères)
-----}

function Tform1.FromDecimal(base : integer):string;
var  Reste      : integer;           // Reste de la division par la base
var  Nombre     : integer;           // Nombre à convertir
begin
  Nombre := StrToInt(EditDecimal.Text);
  while Nombre > 0 do                 // conversion
  begin
    Reste := Nombre mod Base;
    if Reste > 9 then
      result := chr(integer('A') + integer(Reste) - 10) + result
    else
      result := IntToStr(Nombre mod Base) + result;
    Nombre := Nombre div Base;
  end;
end;

{===== Tform1.ToDecimal =====}
Description : Converti d'une base quelconque en décimal.
               Le resultat est affiché dans EditDecimal
Paramètre 1: Base  => La base d'origine en numérique
Paramètre 2: StrIn => Le nombre à convertir sous forme de chaîne de caractères
-----}

procedure Tform1.ToDecimal(base : integer; StrIn : string);
var resultat,
    multiplie,
    tempo,
    i           : integer;           // Il sera affiché dans EditDecimal
                                         // Puissance de la base
                                         // Sauvegarde temporaire
                                         // Indice du caractère
begin
  resultat := 0;
  multiplie := 1;
  for i := length(StrIn) downto 1 do   // conversion
  begin
    if StrIn[i] in ['A'..'F'] then
      tempo := integer(StrIn[i]) - integer('A') + 10
    else
      tempo := StrToInt(StrIn[i]);
    resultat := resultat + (tempo * multiplie);
    multiplie := multiplie * base;
  end;
  EditDecimal.Text := IntToStr(resultat);
end;

{===== Tform1.EditBinKeyUp =====}
Description : Test le nombre et les caractères entré puis converti de la base 2
               dans les autres bases.
-----}

procedure Tform1.EditBinKeyUp(Sender: TObject; var Key: Word; Shift: TShiftState);
begin
  if Length(EditBin.Text) > 0 then
  begin
    ToDecimal(BASE_2, EditBin.Text);
    EditOct.Text := FromDecimal(BASE_8);
    EditHexa.Text := FromDecimal(BASE_16);
  end
  else
    EffaceTout();
end;
```



```

//===== Test si le caractère est "binaire" =====
procedure TForm1.EditBinKeyPress(Sender: TObject; var Key: Char);
begin
    if not (Key in ['0'..'1',BACK_SPACE]) then
        Key := NUL;
end;

//===== Test si le caractère est "octal" =====
procedure TForm1.EditOctKeyPress(Sender: TObject; var Key: Char);
begin
    if not (Key in ['0'..'7',BACK_SPACE]) then
        Key := NUL;
end;

//===== Test si le caractère est "décimal" =====
procedure TForm1.EditDecimalKeyPress(Sender: TObject; var Key: Char);
begin
    if not (Key in ['0'..'9',BACK_SPACE]) then
        Key := NUL;
end;

//===== Test si le caractère est "hexadécimal" =====
procedure TForm1.EditHexaKeyPress(Sender: TObject; var Key: Char);
begin
    Key := upcase(Key);
    if not (Key in ['0'..'9','A'..'F',BACK_SPACE]) then
        Key := NUL;
end;

{
Description : Test le nombre puis converti de la base 8 dans les autres bases.
}
procedure TForm1.EditOctKeyUp(Sender: TObject; var Key: Word;Shift: TShiftState);
begin
    if Length(EditOct.Text) > 0 then
        begin
            ToDecimal(BASE_8,EditOct.Text);
            EditBin.Text := FromDecimal(BASE_2);
            EditHexa.Text := FromDecimal(BASE_16);
        end
    else
        EffaceTout();
end;

{
Description : Test le nombre puis converti de la base 16 dans les autres bases.
}
procedure TForm1.EditHexaKeyUp(Sender: TObject;var Key: Word;Shift: TShiftState);
begin
    if Length(EditHexa.Text) > 0 then
        begin
            ToDecimal(BASE_16,EditHexa.Text);
            EditBin.Text := FromDecimal(BASE_2);
            EditOct.Text := FromDecimal(BASE_8);
        end
    else
        EffaceTout();
end;

{
Description : Test le nombre puis converti de la base 10 dans les autres bases.
}
procedure TForm1.EditDecimalKeyUp(Sender: TObject; var Key: Word;
                                   Shift: TShiftState);
begin
    if Length(EditDecimal.Text) > 0 then
        begin
            EditBin.Text := FromDecimal(BASE_2);
            EditOct.Text := FromDecimal(BASE_8);
            EditHexa.Text := FromDecimal(BASE_16);
        end
    else
        EffaceTout();
end;

//===== end =====
end.

```