



**BRANCHE :** Informatique – Partie écrite

**DATE :** 23 mai 2012

**DURÉE :** 90 min.

**1. Algorithme connu: Equation du 2<sup>nd</sup> degré (10 p.)**

Dessinez le structogramme pour déterminer la ou les solutions d'une équation du premier ou du second degré du type  $ax^2 + bx + c = 0$  (séparation des cas pour  $a, b$  ou  $c=0$ ) ! Les coefficients  $a, b$  et  $c$  (nombres réels) sont entrés séparément au clavier dans trois boîtes d'édition. Les solutions sont affichées dans des étiquettes. Pour chaque cas, un message explicite est affiché à l'aide d'une étiquette.

**2. Algorithme inconnu: Consultation d'un dictionnaire (10 p.)**

a) Donnez le code Delphi/Pascal de la procédure indépendante EXTRAIRE qui permet d'extraire, à partir d'une chaîne de caractères, sa partie gauche et sa partie droite. Les deux parties sont séparées par un point-virgule. La chaîne de caractères CH, sa partie gauche GAUCHE et sa partie droite DROITE sont des paramètres de la procédure !

Exemple d'un appel de la procédure :

EXTRAIRE('pain;bread', G, D); → G obtient la valeur 'pain' et  
D obtient la valeur 'bread'. (3 p.)

b) Soit l'application de consultation d'un dictionnaire montrée par les **figures 1-3**. Les mots du dictionnaire se trouvent dans la liste **lbDict**. Les éléments de la liste sont sous la forme **gauche;droite** où la partie gauche désigne un mot dans la première langue et la partie droite sa traduction dans la deuxième langue. Le premier élément de la liste indique de quelles langues il s'agit (dans l'exemple montré: 'Français;Anglais'). Lorsque l'utilisateur introduit une ou plusieurs lettres dans la boîte **edtMot**, tous les mots commençant par ces lettres ainsi que leur traduction sont affichés dans la grille **sgDict**.

Dessinez le structogramme de la procédure de réaction **edtMotChange** qui effectue le traitement suivant lorsque le contenu de **edtMot** est modifié :

- les langues du dictionnaire sont écrites dans la première ligne (ligne figée) de la grille **sgDict** ;
- toutes les entrées du dictionnaire de la liste **lbDict** dont la partie gauche commence avec les lettres contenus dans **edtMot** sont transférées dans la grille **sgDict** comme le montre la **figure 2** ;
- le nombre de lignes de la grille doit correspondre aux nombre d'entrées trouvées ; le transfert doit être réalisé de manière à ce que la dernière ligne de la grille **sgDict** soit toujours vide (voir **figures 2 et 3**)! Ainsi, si aucune entrée n'est trouvée, la grille affiche la ligne figée avec le nom des langues et une ligne vide (voir **figure 3**).

**Remarque :** Il est recommandé de faire appel à la procédure EXTRAIRE, développée au point a), pour réaliser la procédure de réaction **edtMotChange** ! (7 p.)



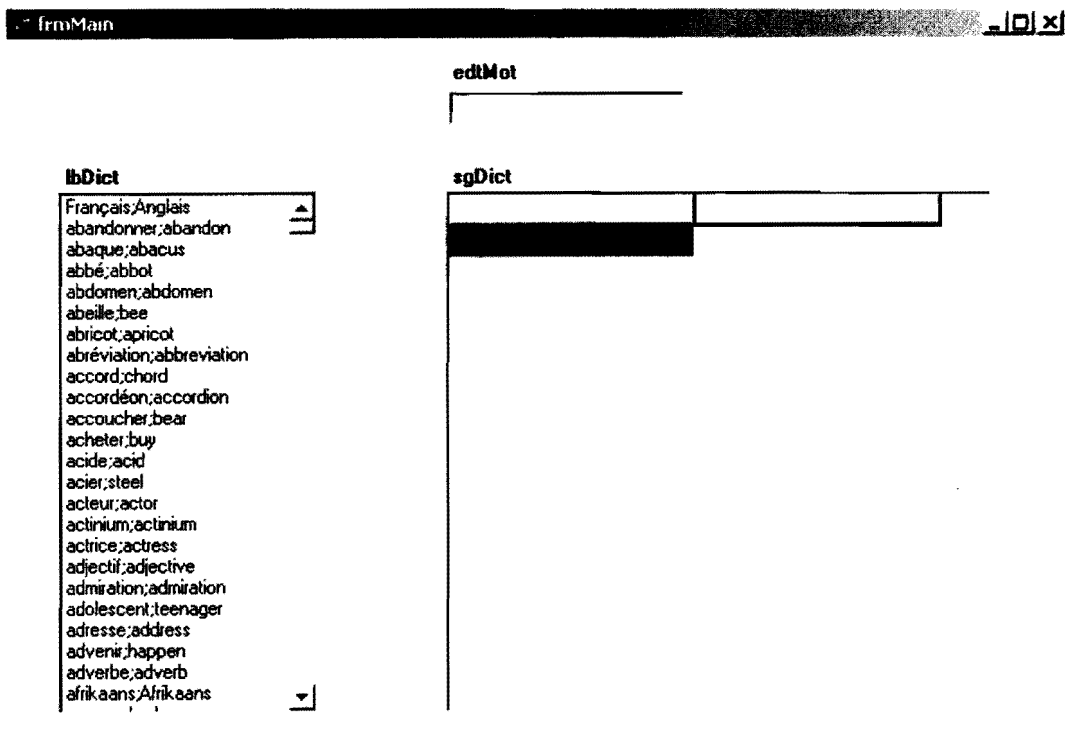
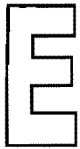


Figure 1 : État de la grille sgDict au lancement du programme

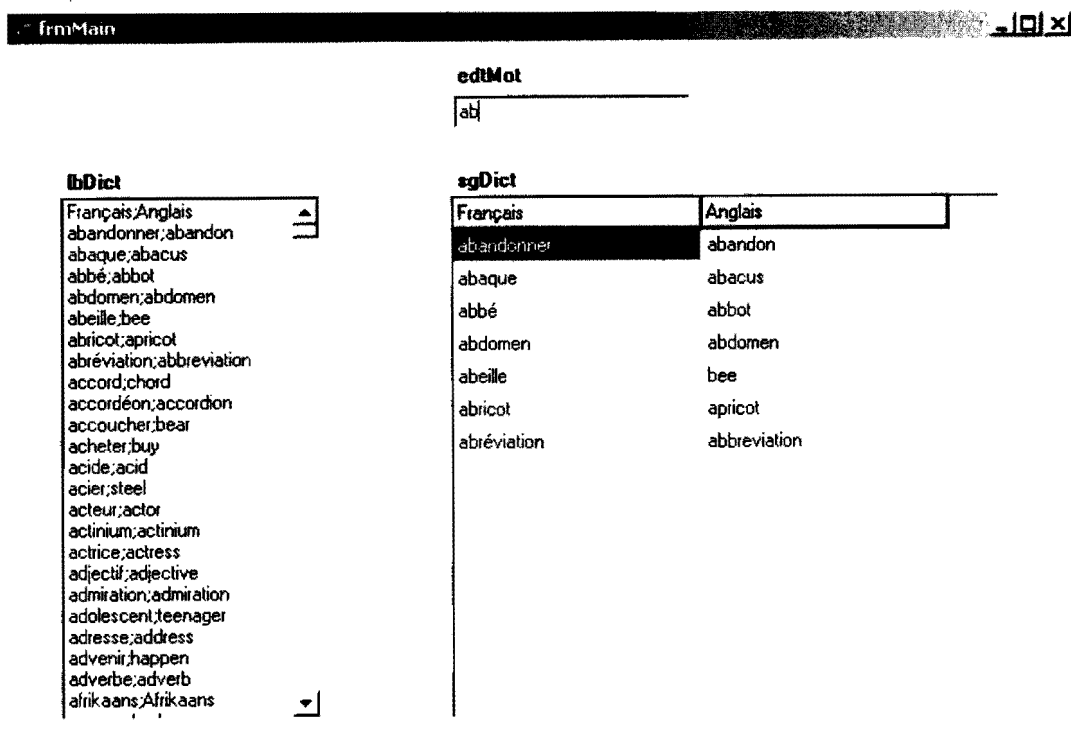
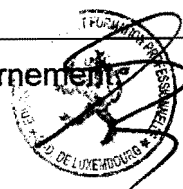
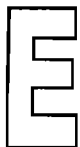


Figure 2 : État de la grille sgDict après l'entrée des lettres 'ab' dans edtMot





Ministère de l'Éducation nationale et de la Formation professionnelle  
EXAMEN DE FIN D'ÉTUDES SECONDAIRES TECHNIQUES  
Régime technique – Division technique générale  
Session 2012

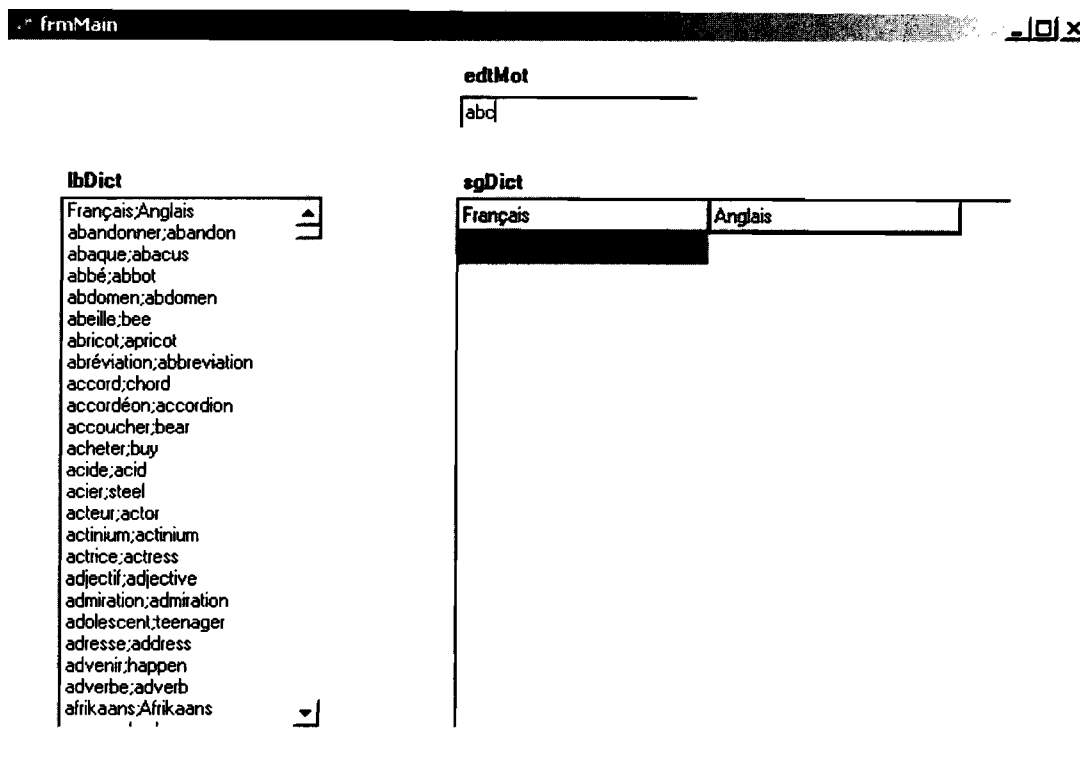


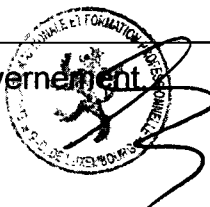
Figure 3 : Etat de la grille sgDict si aucun mot n'est trouvé

### 3. Modification d'un programme (10 p.)

Soit la procédure de réaction `btnTransfererClick` qui transfère les caractères minuscules de la chaîne contenue dans la boîte d'édition `edtChaine` vers la boîte d'édition `edtMinuscules` et les caractères majuscules vers la boîte d'édition `edtMajuscules`.

```
procedure TfrmMain.btnTransfererClick(Sender: TObject);
var CH, MINUS, MAJUS, C:String;
    I:Integer;
begin
    CH:=edtChaine.Text;
    MINUS:=''; MAJUS:='';
    for I:=1 to length(CH) do
        begin
            C:=copy(CH, I, 1);
            if (C>='a') and (C<='z') then MINUS:=MINUS+C;
            if (C>='A') and (C<='Z') then MAJUS:=MAJUS+C;
        end;
    edtMinuscules.Text:=MINUS;
    edtMajuscules.Text:=MAJUS;
end;
```

La figure 4 montre le fonctionnement du bouton `btnTransferer`.



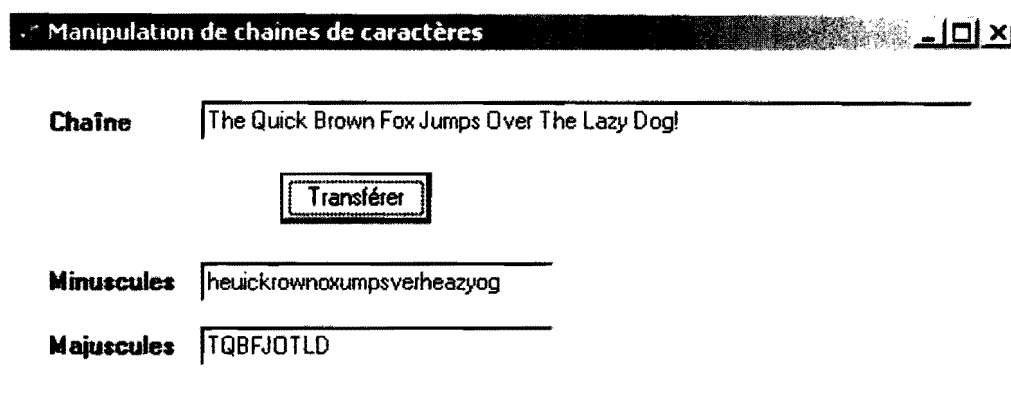
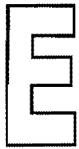


Figure 4 : Fonctionnement du bouton btnTransferer

- a) Modifiez la procédure de réaction **btnTransfererClick** de manière qu'elle fasse appel à une procédure indépendante MINMAJ qui permet de copier, à partir d'une chaîne CH, les caractères minuscules dans la chaîne MINUS et les caractères majuscules dans la chaîne MAJUS. CH, MINUS et MAJUS sont des paramètres de la procédure. Écrivez le code Delphi/Pascal de la procédure MINMAJ et de la procédure de réaction btnTransfererClick modifiée ! (2+2 p.)
- b) Modifiez la méthode **btnTransfererClick** de manière qu'elle fasse appel à deux fonctions indépendantes MINUS et MAJUS qui retournent une chaîne contenant les caractères minuscules respectivement majuscules d'une chaîne de caractères passée comme paramètre. Écrivez le code Delphi/Pascal des deux fonctions MINUS et MAJUS et de la procédure de réaction btnTransfererClick modifiée ! (2+2+2 p.)



## Commission Nationale pour les Programmes d'Informatique

Liste des composants, propriétés, événements et méthodes à connaître pour l'épreuve en informatique à l'examen de fin d'études secondaires techniques - division technique générale

Composant	Préfixe	Propriétés	Evénements	Méthodes
<i>Tous les composants visuels</i>		Name, Width, Height, Top, Left, Enabled, Visible		
<i>Certains composants visuels</i>		Hint, ShowHint, Font, Color, Alignment, Align	OnClick	SetFocus
TForm	<i>frm</i>	Caption, ClientWidth, ClientHeight	OnCreate, OnClose	Close
TButton	<i>btn</i>	Caption		
TEdit	<i>edt</i>	Text	OnChange	Clear, SelectAll
TLabel	<i>lbl</i>	Caption		
TPanel	<i>pnl</i>	Caption		
TImage	<i>img</i>	Picture, Autosize, Canvas	OnMouseDown, OnMouseMove, OnMouseUp	Picture.LoadFromFile, Picture.SaveToFile
TTimer	<i>tm</i>	Interval	OnTimer	
TSpinEdit	<i>spe</i>	Value, MaxValue, MinValue, Increment	OnChange	
TMainMenu	<i>mm</i>			
TMenuItem	<i>mmi</i>	Caption, Checked	OnClick	
TOpenDialog, TSaveDialog	<i>dlg</i>	FileName, Filter, DefaultExt		Execute
TFontDialog	<i>dlg</i>	Font		Execute
TColorDialog	<i>dlg</i>	Color		Execute
TStrings (applicable à Items)	-	Count, Strings[ ]		Clear, Append, Insert, Delete, LoadFromFile, SaveToFile
TListBox	<i>lb</i>	Items (voir TStrings), ItemIndex, Sorted		
TRadioGroup	<i>rg</i>	Items (voir TStrings), ItemIndex, Sorted		
TComboBox	<i>cb</i>	Items (voir TStrings), Text, ItemIndex, Sorted	OnChange	
TCanvas	-	Pen.Color, Pen.Width, Brush.Color, Pixels[ ]		LineTo, MoveTo, Rectangle, Ellipse, TextOut
TPaintBox	<i>pb</i>	Canvas	OnMouseDown, OnMouseMove, OnMouseUp, OnPaint	RePaint
TStringGrid	<i>sg</i>	Cells[], Col, Row, FixedCols, FixedRows, RowCount, ColCount, Option - goEditing		

### Fonctions / Procédures

Manipulation de chaînes de caractères	Copy, Length, Pos, Delete, Insert, + (concaténation), LowerCase, UpperCase
Conversion de types	IntToStr, FloatToStr, StrToInt, StrToFloat
Fonctions mathématiques	Abs, Sin, Cos, Exp, Ln, Sqr, Sqrt, Round, Trunc, Random
Affichage de messages	ShowMessage

### Syntaxe de fonctions et procédures agissant sur des chaînes de caractères

Copy(<chaîne>, <position début>, <nombre de caractères>)
Pos(<sous-chaîne>, <chaîne>)
Delete(<chaîne>, <position début>, <nombre de caractères>)
Insert(<sous-chaîne>, <chaîne>, <position d'insertion>)

### Valeurs de paramètres d'événements

Shift: <i>ssShift</i> , <i>ssCtrl</i> , <i>ssAlt</i> , <i>ssLeft</i> , <i>ssRight</i> , <i>ssMiddle</i> , <i>ssDouble</i>
Button: <i>mbLeft</i> , <i>mbRight</i> , <i>mbMiddle</i>

