Code branche	Ministère de l'Education nationale et de la Formation professionnelle EXAMEN DE FIN D'ETUDES SECONDAIRES TECHNIQUES Régime technique – Division technique générale **Section technique générale - Session 2012/2013**	
INFOR D Th		
Épreuve **écrite**	*Branche*	*Division / Section*
Durée épreuve **1h30**	**Informatique** **(Delphi – partie théorique)**	**GE**
Date épreuve 3.6.2013		

1. **Algorithme connu** (7 p.)

   Écrire pour le problème suivant l'algorithme sous forme de structogramme :

   Une phrase est formée par des mots séparés par un certain caractère. Transférer tous les mots de cette phrase un par un dans une liste du type *TListBox*. La phrase et le caractère séparateur sont entrés à l'aide des champs **edtChaineInitiale** respectivement **edtSeparateur** (du type *TEditBox*). On peut supposer que la phrase commence et se termine par un mot (pas de caractère séparateur ni de point) et qu'elle ne contient pas de caractères séparateurs consécutifs.

---

2. **Algorithme inconnu** (4 + 4 + 4 = 12 p.)

   Réaliser une application qui permet d'écrire les diviseurs d'une suite de nombres dans une grille et de transférer les nombres premiers de cette suite dans une liste.

   L'application contient la procédure **COL_DIVISEURS**, la méthode **btnDiviseursClick** et la méthode **btnPremiersClick**. Ces trois sous-programmes sont à décrire à l'aide de structogrammes.

   - La procédure indépendante **COL_DIVISEURS** accepte une grille **sgDiv** (*TStringGrid*) et un entier **C** comme paramètres. Elle détermine les diviseurs du nombre qui se trouve dans la cellule supérieure de la colonne d'indice **C** de la grille **sgDiv**. Ces diviseurs sont écrits un à un dans les cellules en-dessous du nombre. Une rangée est ajoutée à la grille uniquement en cas de besoin c.-à.-d. s'il n'y a plus de cellule libre pour contenir un diviseur.

     Exemple (extrait de sgDiv ; diviseurs du nombre 12) :

	colonne d'indice C
rangée d'indice 0	12
1	1
2	2
3	3
4	4
5	6
6	12

- La méthode **btnDiviseursClick** remplit d'abord la rangée supérieure d'une grille **sgDiv** avec les nombres entiers compris entre deux nombres donnés (limites comprises). Ces deux limites sont entrées à travers deux boîtes **speN1** et **speN2** (*TSpinEdit*). Ensuite, pour chacun de ces nombres, les diviseurs sont déterminés et écrits dans les cellules en-dessous.

  Remarques :
  - Le nombre de lignes et de colonnes de la grille sera adapté à la quantité de nombres et de diviseurs (voir exemple).
  - On peut supposer qu'au départ la grille est vide.
  - La procédure COL_DIVISEURS est appelée à l'intérieur de cette méthode.

  Exemple :

10	11	12	13	14	15	16	17
1	1	1	1	1	1	1	1
2	11	2	13	2	3	2	17
5		3		7	5	4	
10		4		14	15	8	
		6				16	
		12					

Premier nombre : 10 Dernier Nombre : 17

- La méthode **btnPremiersClick** réalise le transfert vers une liste **lbPremiers** des nombres premiers parmi les nombres de la première ligne de la grille **sgDiv**.

  Remarques :
  - Un nombre est premier s'il a exactement deux diviseurs (1 et lui-même). Ainsi, 11 est premier, tandis que 1 (un seul diviseur) et 10 (quatre diviseurs) ne sont pas premiers.
  - Avant le transfert la liste est effacée.

  Exemple : Parmi les nombres de l'exemple précédent les nombres 11, 13 et 17 sont transférés vers la liste.

---

3. **Modification d'un programme** (3 + 3 = 6 p.)

Soit la méthode suivante qui détermine si un caractère donné se trouve exactement deux fois dans une chaîne donnée.

```
procedure TfrmMain.btnTestClick(Sender: TObject);
var I, COMPT : integer;
    CH, C : string;
begin
   CH := edtChaine.Text;
   C := edtCar.Text;
   COMPT := 0;
   for I:= 1 to length(CH) do
      if copy(CH,I,1)=C then COMPT := COMPT+1;
   if COMPT=2
      then lblMessage.Caption := 'Le caractère apparaît exactement 2 fois'
      else lblMessage.Caption := 'Le caractère n''apparaît pas exactement 2 fois'
end;
```

(a) Modifier la méthode de manière qu'elle fasse appel à une fonction OCC2_BOOLEAN qui accepte une chaîne et un caractère (du type *String*) comme paramètres et qui retourne une valeur booléenne, à savoir `true` si la chaîne contient le caractère exactement 2 fois et `false` dans le cas contraire. Ecrire le code Delphi de la fonction OCC2_BOOLEAN et de la méthode modifiée.

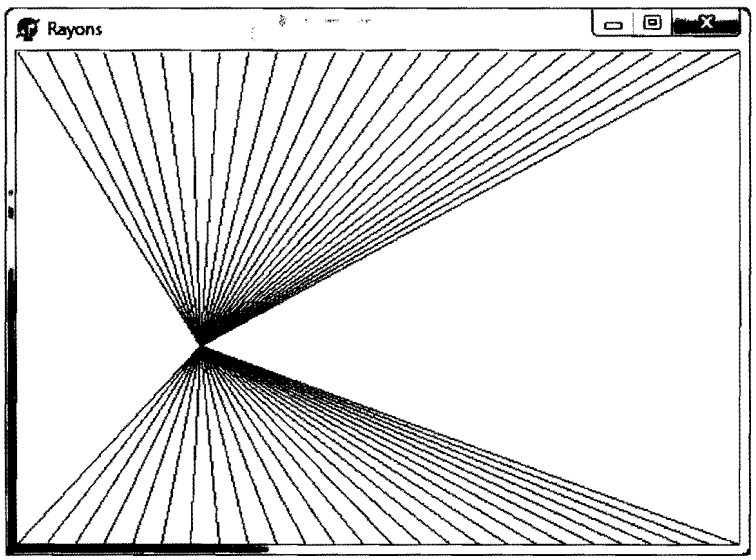
(b) Modifier la méthode de manière qu'elle fasse appel à une fonction OCC2_STRING qui accepte une chaîne et un caractère (du type *String*) comme paramètres et qui retourne une chaîne de caractères (du type *String*) appropriée. Ecrire le code Delphi de la fonction OCC2_BOOLEAN et de la méthode modifiée.

---

4. **Question de compréhension** (5 p.)

La méthode ci-dessous est sensée réaliser un dessin comme celui de la figure au moment où on enfonce le bouton droit de la souris. Le point commun de toutes les lignes est l'endroit où on a activé le bouton de la souris. Toutes les lignes sont rouges. Le dessin se fait sur une boîte à peindre **pbLignes**. Toutefois la méthode contient plusieurs erreurs (de syntaxe respectivement de logique).

Décrire ou corriger ces erreurs.




```
procedure TfrmMain.pbLignesMouseDown(Sender: TObject; Button:
          TMouseButton; Shift: TShiftState; X, Y: Integer);
var I : real;
begin
   if Button = ssRight then
      begin
         I:=0;
         pbLignes.Canvas.Color := clRed;
         while I = pbLignes.Width do
            begin
               pbLignes.Canvas.LineTo(X,Y,I,0);
               pbLignes.Canvas.LineTo(X,Y,I,pbLignes.Height);
               I := I+20
            end
       end
end;
```

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
Tous les composants visuels		Name, Width, Height, Top, Left, Enabled, Visible		
Certains composants visuels		Hint, ShowHint, Font, Color, Alignment, Align	OnClick	SetFocus
TForm	*frm*	Caption, ClientWidth, ClientHeight	OnCreate, OnClose	Close
TButton	*btn*	Caption		
TEdit	*edt*	Text	OnChange	Clear, SelectAll
TLabel	*lbl*	Caption		
TPanel	*pnl*	Caption		
TImage	*img*	Picture, Autosize, Canvas	OnMouseDown, OnMouseMove, OnMouseUp	Picture.LoadFromFile, Picture.SaveToFile
TTimer	*tm*	Interval	OnTimer	
TSpinEdit	*spe*	Value, MaxValue, MinValue, Increment	OnChange	
TMainMenu	*mm*			
TMenuItem	*mmi*	Caption, Checked	OnClick	
TOpenDialog, TSaveDialog	*dlg*	FileName, Filter, DefaultExt		Execute
TFontDialog	*dlg*	Font		Execute
TColorDialog	*dlg*	Color		Execute
TStrings (applicable à Items)	-	Count, Strings[ ]		Clear, Append, Insert, Delete, LoadFromFile, SaveToFile
TListBox	*lb*	Items (voir TStrings), ItemIndex, Sorted		
TRadioGroup	*rg*	Items (voir TStrings), ItemIndex, Sorted		
TComboBox	*cb*	Items (voir TStrings), Text, ItemIndex, Sorted	OnChange	
TCanvas	-	Pen.Color, Pen.Width, Brush.Color, Pixels[ ]		LineTo, MoveTo, Rectangle, Ellipse, TextOut
TPaintBox	*pb*	Canvas	OnMouseDown, OnMouseMove, OnMouseUp, OnPaint	RePaint
TStringGrid	*sg*	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: *ssShift, ssCtrl, ssAlt, ssLeft, ssRight, ssMiddle, ssDouble*
Button: *mbLeft, mbRight, mbMiddle*