

*Durée* : 2h (1h pour l'épreuve écrite et 1h pour l'épreuve pratique.)

### Question 1

16 (=1+10+5) points

- (1) Exposer l'*idée* de l'algorithme de *tri par insertion*.
- (2) Donner une *solution récursive* de cet algorithme.
- (3) On veut trier la liste lbExemple : =('D', 'B', 'C', 'A') à l'aide de la procédure précédente.
  - a) Ecrire l'*appel* de la procédure.
  - b) Quels *changements* la liste lbExemple subit-elle au cours de l'exécution ?

### Question 2

6 points

Corriger les *erreurs logiques et syntaxiques* dans la procédure WithManyBugs ci-dessous. La procédure doit copier les éléments de la liste lbLili dans chaque colonne du tableau carré sgTata.

```
procedure WithManyBugs(lbLili:ListBox; sgTata:StringGrid); // (1)
begin // (2)
  n:=lbLili.Count; // (3)
  sgTata.ColCount:=sgTata.RowCount:=n-1; // (4)
  for i:= 0 to n do // (5)
    for j:= 0 to n do // (6)
      sgTata.Items[i,j]:=lbLili.Items[i]; // (7)
  end; // (8)
```

*Remarque* : Une ligne sans erreurs n'a pas besoin d'être recopiée. Si une ligne supplémentaire devra être insérée entre les lignes (n) et (n+1), cette nouvelle ligne aura le numéro (n').

### Question 3

8 (=2+2+4) points

Voici la fonction MyCalculus :

```
function MyCalculus(a,b:integer):integer;
begin
  if a<0 then Result:=-MyCalculus(-a,b)
  else if a=0 then Result:=0
  else Result:=b+MyCalculus(a-1,b)
end;
```

- (1) Calculer en détaillant MyCalculus(3,7) ;
- (2) Calculer en détaillant MyCalculus(-2,-4) ;
- (3) Que calcule la fonction MyCalculus en général ? Justifier !

#### Question 4 (avec ordinateur)

30 (=8+8+10+4) points

1	2	3	4	5	6	7
8	X	10	11	12	13	14
15	16	17	X	19	20	21
22	23	24	25	26	X	X
29	30	31	32	33	34	35
36	37	38	X	40	41	42
43	44	45	46	47	X	49

Au loto, le joueur dispose d'une grille 7x7 contenant les entiers de 1 à 49 comme sur la figure ci-contre. Il choisit 6 entiers distincts en marquant les cases correspondantes d'une croix. Le but de cette question est de programmer une simulation partielle du jeu dans Delphi.

- (1) Créer une interface graphique avec les éléments suivants :
  - une grille 7x7 de type **TStringGrid**, nommée **sgLoto**, vide au lancement de l'application ;
  - une liste, **lbTirage**, également vide au lancement de l'application ;
  - deux boutons avec les « captions » **Initialiser** et **Tirage**.
- (2) Ecrire une procédure **FillGrid** qui remplit la grille **sgLoto** avec les entiers de 1 à 49, comme sur la figure.
- (3) Ecrire une procédure **GenerateRandomList** qui remplit la liste **lbTirage** avec 6 entiers distincts choisis au hasard entre 1 et 49.
- (4) Déroulement du jeu :
  - Un clic sur le bouton **Initialiser** aura comme effet d'appeler la procédure **FillGrid** et de vider la liste **lbTirage**.
  - Le joueur choisit ses numéros en écrivant la lettre '**X**' dans 6 cases de la grille. (Bien sûr les entiers correspondants disparaissent alors !)
  - Un clic sur le bouton **Tirage** devra appeler la procédure **GenerateRandomList**.
  - Dans cette version simplifiée du jeu, le joueur devra chercher lui-même les numéros gagnants.

G. Lorang et R. Weber