

Question 1

16 (=1+10+5) points

- (1) Voir cours.
- (2) Voir cours.
- (3) lbExemple := ('D', 'B', 'C', 'A').
 - a) tri_insertion_r(lbExemple, 0, 3);
 - b) ('D', 'B', 'C', 'A') (état initial)
 - ('D', 'D', 'C', 'A')
 - ('B', 'D', 'C', 'A')
 - ('B', 'D', 'D', 'A')
 - ('B', 'C', 'D', 'A')
 - ('B', 'C', 'D', 'D')
 - ('B', 'C', 'C', 'D')
 - ('B', 'B', 'C', 'D')
 - ('A', 'B', 'C', 'D') (état final)

Question 2

6 points

```

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

```

Question 3

8 (=2+2+3) points

- (1) $\text{MyCalculus}(3,7) = 7 + \text{MyCalculus}(2,7)$
 $= 7 + 7 + \text{MyCalculus}(1,7)$
 $= 7 + 7 + 7 + \text{MyCalculus}(0,7)$
 $= 7 + 7 + 7 + 0$
 $= 21$
- (2) $\text{MyCalculus}(-2,-4) = - \text{MyCalculus}(2,-4)$
 $= -(-4 + \text{MyCalculus}(1,-4))$
 $= -(-4 + (-4) + \text{MyCalculus}(0,-4))$
 $= -(-4 + (-4) + 0)$
 $= -(-8) = 8$

(3) En général : $\text{MyCalculus}(a, b) = a \cdot b$.

En effet, montrons cette thèse d'abord pour $a \geq 0$, par récurrence.

Si $a = 0$ alors la thèse est vraie car $\text{MyCalculus}(0, b) = 0$.

Supposons le résultat vrai jusqu'au rang $a - 1$ et montrons qu'il reste vrai au rang suivant :

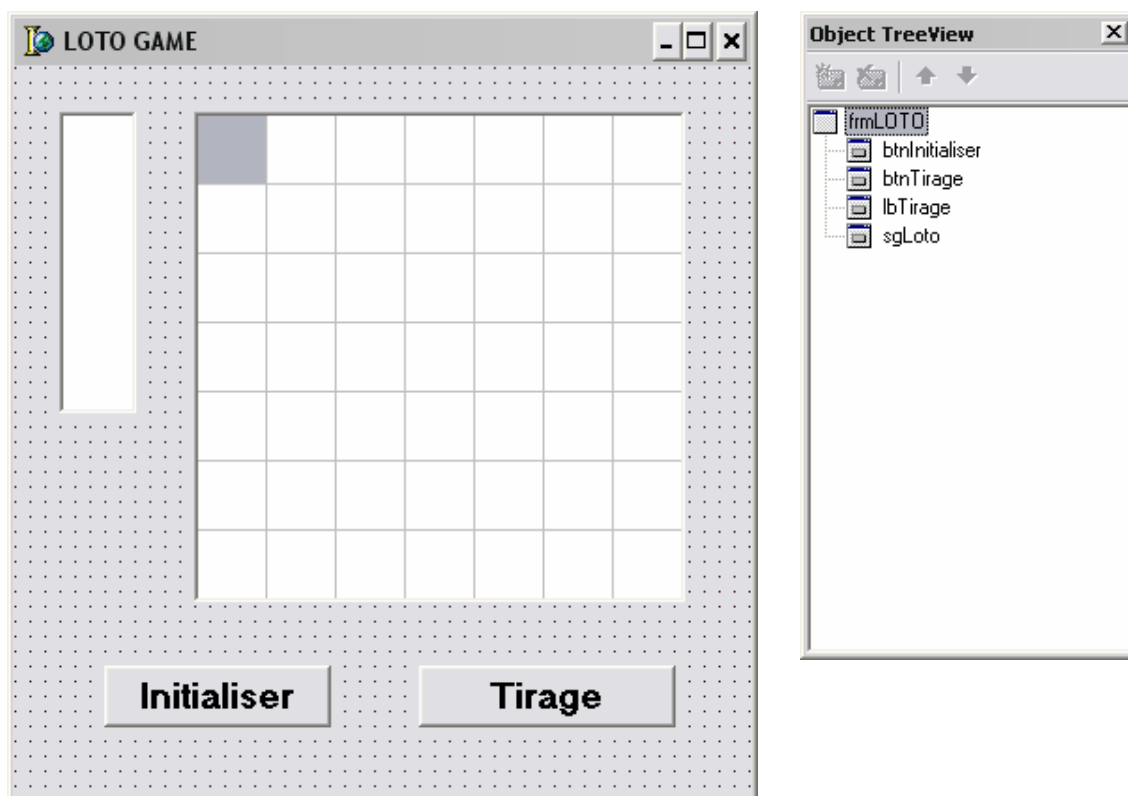
$$\begin{aligned}\text{MyCalculus}(a, b) &= b + \text{MyCalculus}(a-1, b) \\ &= b + (a-1) \cdot b \text{ (hyp. de réc.)} \\ &= ab\end{aligned}$$

Il reste à voir que la thèse est vraie pour $a < 0$. Or, dans ce cas :

$$\begin{aligned}\text{MyCalculus}(a, b) &= - \text{MyCalculus}(-a, b) \\ &= -(-a) \cdot b \text{ (thèse vraie car } -a > 0) \\ &= ab\end{aligned}$$

Question 4

(1)



Attention : l'option *goEditing* du `sgLoto` doit être mise à « true » !

(2)

```
Procedure FillGrid(Table: TStringGrid);  
var i, j, k: integer;  
begin  
    k:=1;  
    for i:=0 to Table.RowCount-1 do  
        for j:=0 to Table.ColCount-1 do  
            begin  
                Table.Cells[j,i]:=inttostr(k);  
                k:=k+1;  
            end  
    end;
```

(3)

```
procedure GenerateRandomList(liste: Tlistbox);  
var  
    i, n: integer;  
    flag: boolean;  
begin  
    randomize;  
    liste.Clear;  
    while liste.Count < 6 do  
        begin  
            n:=1+Random(49);  
            flag:=false;  
            for i:=0 to liste.Count-1 do  
                if strtoint(liste.Items[i])=n then flag:=true;  
                if flag = false then liste.Items.Append(inttostr(n));  
            end;  
    end;
```

(4)

```
procedure TfrmLOTO.btnInitialiserClick(Sender: TObject);  
begin  
    FillGrid(sgLoto);  
    lbTirage.Clear;  
end;  
  
procedure TfrmLOTO.btnTirageClick(Sender: TObject);  
begin  
    GenerateRandomList (lbTirage);  
end;
```

G. Lorang et R. Weber