

Devoir en classe d'Informatique II,2

50 minutes – 30 points

Exercice 1

[30 p.]

1. Présentez l'algorithme du *Tri rapide*:

a. La procédure *quicksort*

[4 p.]

```
procedure quicksort(list : TListBox; left, right : integer);
var
  pivotIndex, newPivotIndex : integer;
begin
  if (right > left) then
    begin
      pivotIndex := right;
      newPivotIndex := division(list, left, right, pivotIndex);
      quicksort(list, left, newPivotIndex-1);
      quicksort(list, newPivotIndex+1, right);
    end;
  end;
```

b. La fonction auxiliaire *division*

[6 p.]

```
function division(list: TListbox; left, right, pivotIndex : integer) : integer;
var
  pivotValue : String;
  storeIndex, i : integer;
begin
  pivotValue := list.Items[pivotIndex];
  swap(list, pivotIndex, right);
  storeIndex := left;
  for i := left to right - 1 do
    if (list.Items[i] <= pivotValue) then
      begin
        swap(list, i, storeIndex);
        storeIndex := storeIndex + 1;
      end;
  swap(list, storeIndex, right);
  result := storeIndex;
end;
```

c. La procédure auxiliaire *swap*

[2 p.]

```
procedure swap(list : TListBox; i, j : integer);
var
  tmp : string;
begin
  tmp := list.Items[i];
  list.Items[i] := list.Items[j];
  list.Items[j] := tmp;
end;
```

2. Donnez un exemple d'exécution de l'appel « *division(lbList, 0, 7, 7)* » en utilisant une TListBox, appelée « lbList », contenant les valeurs (de type **string**) suivantes: [8 p.]

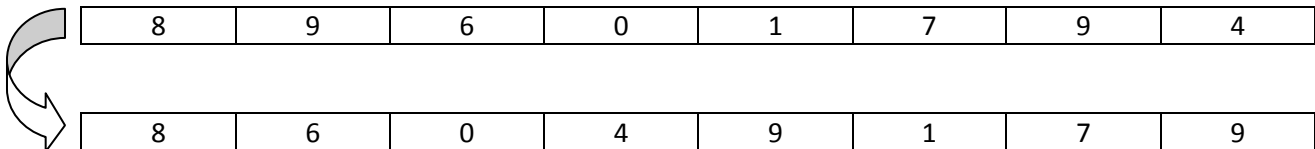
4	7	1	3	12	5	4	9
---	---	---	---	----	---	---	---

Les éléments de la listbox ne changent pas de position. Veuillez-vous référer à la notation sur cours sur les exemple d'exécution de l'algorithme tri rapide.

3. En vous basant sur le fonctionnement de la fonction *division*, écrivez une fonction *division2* qui « trie » une listbox, contenant des nombres naturels, de manière à ce que tous les nombres pairs se trouvent au début et tous les nombres impairs se trouvent à la fin. [8 p.]

Remarque : L'ordre relatif des nombres pairs ou impairs n'est pas à considérer !

Exemple :



```

function division2(list: TListbox; left, right, pivotIndex : integer) : integer;
var
  pivotValue : String;
  storeIndex, i : integer;
begin
  {pivotValue := list.Items[pivotIndex];
  swap(list, pivotIndex, right);}
  storeIndex := left;
  for i := left to right - 1 do
    if (StrToInt(list.Items[i]) mod 2 = 0) then
      begin
        swap(list, i, storeIndex);
        storeIndex := storeIndex + 1;
      end;
  {swap(list, storeIndex, right);}
  result := storeIndex;
end;

```

4. Quel autre algorithme de tri obtient-on, si on remplace le choix du pivot dans la procédure *quicksort* par un appel à une fonction, qui retourne la position de la valeur maximale dans le sous-ensemble [*left, right*] ? [2 p.]

On obtient l'algorithme du tri par sélection. Lors de chaque appel récursif sur un ensemble de n éléments, seulement le pivot retrouve sa position finale. Or, comme le pivot est la valeur maximale, seulement la valeur maximale retrouve sa position finale et l'algorithme effectue un appel récursif sur un ensemble de $n-1$ éléments. Tout comme l'algorithme du tri par sélection, on a donc un ensemble de nombres triés qui ont atteint leur place finale et auxquelles on rajoute un élément par appel récursif.