Correction of the Tutorial Series N°2

Exercise 1:

1. Procedure creating a file and recording in this file the integers from 1 to **n**:

```
Procedure createFile(n:integer);
Var f: File;  i: Integer;
Begin
Assign(f, "C:/Numbers.txt");
Open(f, "w");
For i ← 1 To n Do
    FWriteLn(f, i);
Close(f);
End.
```

2. Function returning the sum of integers contained in a file:

```
Function calculateSum(path:String):integer;
Var f: File; x,s: Integer;
Begin
Assign(f, path);
Open(f, "r");
s ← 0;
While EOF(f) = False
  Begin
  FRead(f,x);
  s ← s + x;
  End;
Close(f);
calculateSum ← s;
End.
```

Exercise 2:

Function returning the total number of words in a file:

```
function countWords(path:String):integer;
Var f: File; i,nbWordsInLine,nbWordsTot: Integer;line:String;
Begin
Assign(f, path);
Open(f, "r");
nbWordsTot ← 0;
While EOF(f) = False
  Begin
  FReadLn(f,line);
  i ← 1;
  nbWordsInLine ← 1;
  While i ≤ Length(line) Do
```

```
        Begin
        If line(i)= ' ' Then
             nbWordsInLine ← nbWordsInLine + 1;
        i ← i+1;
        End;
    nbWordsTot ← nbWordsTot + nbWordsInLine;
    End;
Close(f);
countWords ← nbWordsTot;
End.
```

Exercise 3:

Procedure copying the content of a source text file to a destination file:

```
Procedure copyFile(sourcePath,destinationPath:String);
Var source,destination:File; line: String ;
Begin
Assign(source, sourcePath);
Open(source, "r");
Assign(destination, distinationPath);
Open(destination, "w");
While EOF(source) = False
  Begin
  FReadLn(source,line);
  FWriteLn(destination,line);
  End;
Close(source);
Close(destination);
End.
```

Exercise 4:

Procedure returning the average mark and the name of the student with the highest mark from a text file of students:

```
Procedure copyFile(path:String;Var avg:Real; Var name:String);
Var f:File; mark,s,max: real; x: String; nb: integer;
Begin
Assign(f, path);
Open(f, "r");
s ← 0;
nb ← 0;
max ← -1;
While EOF(f) = False
  Begin
  Fread(f,x);
  Fread(f,mark);
  s ← s + mark;
  nb ← nb + 1;
  If mark > max then
       Begin
       max ← mark;
```

```
        name ← x;
        End;
    End;
  Close(f);
  avg ← s/nb;
  End.
```

Exercise 5:

Procedure that merges the content of two text files into a single file:

```
  Procedure mergeFiles(sourcePath1,sourcePath2,destPath:String);
  Var f1,f2,f:File; line: String;
  Begin
  Assign(f1, sourcePath1);
  Open(f1, "r");
  Assign(f2, sourcePath2);
  Open(f2, "r");
  Assign(f, destPath);
  Open(f, "w");
  While EOF(f1) = False
    Begin
    FReadLn(f1,line);
    FWriteLn(f,line);
    End;
  While EOF(f2) = False
    Begin
    FReadLn(f2,line);
    FWriteLn(f,line);
    End;
  Close(f1);
  Close(f2);
  Close(f3);
  End.
```

Exercise 6:

1. The type describing an employee:

```
  Type Employee = Record
                  Begin
                  ID: Integer;
                  FirstName,LastName,Grade: String[20];
                  Salary: Real;
                  End;
```

2. The type describing a set of employees:

```
  Const n = 100;
  Type TabEmployees = Array[n] of Employee;
```

3. Procedure storing employees' information in a file:

```
Procedure storeEmployees(T: Tab, path:String);
Var f: File; i: integer;
Begin
Assign(f, path);
Open(f, "w");
For i ← 0 to n-1 do
  Begin
  FWrite(f,T[i].ID);
  FWrite(f,T[i].FirstName);
  FWrite(f,T[i].LastName);
  FWrite(f,T[i].Grade);
  FWrite(f,T[i].Salary);
  End;
Close(f);
End;
```

4. Procedure that displays the list of employees whose salary is between **A** and **B**:

```
Procedure displayEmployees(path:String, A,B: integer);
Var f: File;ID:Integer;firstName,lastName,Grade:String[20];
    Salary: Real;
Begin
Assign(f, path);
Open(f, "r");
While EOF(f) = False Do
  Begin
  FRead(f,ID);
  FRead(f,FirstName);
  FRead(f,LastName);
  FRead(f,Grade);
  FRead(f,Salary);
  If Salary ≥ A and Salary ≤ B Then
      Write(firstName,LastName);
  End;
Close(f);
End;
```

5. The main algorithm:

```
Algorithm employeeManagement;
……
Var T: Tab; i:Integer;
Begin
For i ← 0 To n-1 Do
   Begin
   Read(T[i].ID,T[i].FirstName,T[i].LastName);
   Read(T[i].Grade,T[i].Salary);
   End;
storeEmployees(T,"E:/Employees.txt");
displayEmployees("E:/Employees.txt",25000,40000);
End.
```