

## **Exercise 1**

Consider the following algorithm:

```
Algorithm Ex01
  F1, F2, F3: File of integers
  X: integer;
Begin //main program
  ASSIGN (F1, "Numbers.bin"
  ASSIGN (F2, "PosNumbers.bin"
  ASSIGN (F3, "NegNumbers.bin"

  OPEN (F1);
  REWRITE (F2);
  REWRITE (F3);

  While ( ! eof (F1)) do
    Read (F1, X)
    If ( X > 0) do
      Write (F2, X);
    Else
      Write (F3, X);
    End if
  End while

  Close (F1);
  Close (F2);
  Close (F3);
END
```

1. Comment this algorithm? What does this algorithm do?
2. Modify this algorithm to allow it to display the size of the three files considering that an integer is represented using 4 bytes.

## **Exercise 2**

1. Let's consider that we have two textual files: Words.txt and Mots.txt. Write an algorithm that allows determining whether the two files are copies of each other.
2. Modify your algorithm by using a function Check ( file\_name1, file\_name1) that verify whether two files are copies of each other.

## **Exercise 3**

Consider the Student type defined in the previous series. Using the modules defined for the Student type, write the following modules:

1. The procedure Create (file\_name: string, int N) that reads the information of N students and write them in a file identified by its name.
2. The procedure fillVe (file\_name: string, var V: array of integer, var N: integer) that fills an array of students from a file identified by its name.
3. Write a procedure Admitted (file\_name: string) that displays the information of admitted students.

# SOLUTION

## Exercise 1

1. This algorithm splits the file "Numbers.bin" into two files "PosNumbers.bin" containing its positive numbers of and "NegNumbers.bin" containing its negative numbers.
2. Comments + modifications.

### Algorithm Ex01

```
F1, F2, F3: File of integers // declaration of three files of integers
X: integer;
Nb1, Nb2, Nb3: integer
Begin //main program
  /* Assignment of the three files F1, F2, F3 (Algorithmic variables) with three
  binary files "Numbers.bin", "PosNumbers.bin" and "NegNumbers.bin"*/

  ASSIGN (F1, "Numbers.bin");
  ASSIGN (F2, "PosNumbers.bin");
  ASSIGN (F3, "NegNumbers.bin");

  OPEN (F1);      // Opening the file F1 for reading
  REWRITE (F2);  // Creation an opening the file F2 for writhing
  REWRITE (F3);  // Creation an opening the file F3 for writhing

  Nb1 ← 0; Nb2 ← 0; Nb3 ← 0;
  /* the function eof returns true if we attend the end of the file
  otherwise it returns false*/

  While (! eof (F1)) do //

    Read (F1, X) // read an integer x from the file F1
    Nb1 ← Nb1 + 1;
    If ( X > 0) do
      Write (F2, X); // Write an integer x in the file F1
      Nb2 ← Nb2 + 2;
    Else
      Write (F3, X); // Write an integer x in the file F1
      Nb3 ← Nb1 + 3;
    End if
  End while

  Close (F1);
  Close (F2);
  Close (F3);

  Write ("the size of the file Numbers.bin is : ", 4* Nb1);
  Write ("the size of the file PosNumbers.bin is : ", 4* Nb2);
  Write ("the size of the file NegNumbers.bin is : ", 4* Nb3);
```

**END**

## **Exercise 2:**

3. Let's consider that we have two textual files: Words.txt and Mots.txt. Write an algorithm that allows determining whether the two files are copies of each other.

### **Algorithm Exo2**

```
F1, F2: File of characters // declaration of two files
C1, C2: character
Begin //main program
  ASSIGN (F1, "Words.txt");
  ASSIGN (F2, "Mots.txt");

  OPEN (F1);      // Opening the file F1 for reading
  OPEN (F2);      // Opening the file F1 for reading

  While (! eof (F1) and ! eof (F2)) do //
    Read (F1, C1)
    Read (F1, C2)
    If ( C1 ≠ C2) do
      Return false
    End if
  End while
  if (eof (F1) and eof (F2)) then
    write ("these files are copies of each other" ;
  else
    write ("these files are not copies of each other"
  endif
  Close (F1);
  Close (F2);
```

**END**

4. Modify your algorithm by using a function Check ( file\_name1, file\_name1) that verify whether two files are copies of each other.

### **Algorithm Exo2**

```
Function check (file_name1, filename2: string): Boolean
  F1, F2: File of characters // declaration of two files
  C1, C2: character
Begin
  ASSIGN (F1, file_name1);
  ASSIGN (F2, file_name2);
  OPEN (F1);      // Opening the file F1 for reading
  OPEN (F2);      // Opening the file F1 for reading

  While (! eof (F1) and ! eof (F2)) do //
    Read (F1, C1)
    Read (F2, C2)
    If ( C1 ≠ C2) do
      Return false
    End if
  End while
  if (eof (F1) and eof (F2)) then
    return true ;
  else
    return false;
  endif
```

**END**

**Begin**

```
If (check ("Words.txt", "Mots.txt" ) then  
    write ("these files are copies of each other" ;  
Else  
    write ("these files are not copies of each other"  
Endif
```

**End**