Larbi Ben M'hidi –Oum El Bouaghi– University
Faculty of Exact Sciences and Natural and Life Sciences
Department of Mathematics and Computer Science

Level: 1st year "Computer Science" **Date:** 15/01/2026
Module: Algorithmic and Data Structure 1 **Duration**: 1h30m

Exam n°1 "Typical Correction"

Comprehension questions (6 points)

1. Complete the following expressions.

(0.5 point for each correct and complete response) .

a. An algorithm is a sequence of elementary operations written using a pseudo-code.
b. To control the number of repetitions or iterations in a loop instruction, we can use the techniques: **counter** where the number of repetitions is known or **Boolean condition**.
c. An array / matrix is a data structure, it is characterized by: -
1- Identifier, 2- Size, 3- Data Type
d. The main difference between an array and a record: the array allows us to store several values of the **same type**. Whereas, the record permits grouping in a single entity a set of data of **different types**.
e. When we have a multiple choices on a variable, we can use switch-case instruction; however, if the number of choices is small nested conditional instruction is possible.
f. Give the two main operations that can be performed on string variables **Length** and **concatenation**

Length (str) : it provides the length of the string str . **Concat** (str1, str2) : it provides the string obtained by the concatenation of the two strings str1 and str2

g. What is the primary reason for using a repeat loop rather than a while loop? When the loop iterates at least once.
h. Give the operation that expresses the decrement of a counter. count-- or count=count-1;

2. Define the appropriate custom type (in algorithmic or in C) for the following situations:

(1 point for each correct and complete declaration) .

a. The four seasons; "enumerated type" (**0.5 point)**

In algorithmic : Seasons= (Spring, Summer, Autumn, Winter); (**0.5 point)**
In C : **enum** Seasons {Spring, Summer, Autumn, Winter};

b. Product characterized by code, name, price and quantity; "Record" (**0.5 point)**
(**0.5 point)**

In algorithmic	In C
Type Product=**Record** Code: integer; Name: string; Price: real; Quantity: integer; **EndRecord**	**typedef struct** Product { int Code; char Name[20]; float Price; int Quantity; } Product;

Exercise n°1 (4 points)

Write an algorithm that requests a time in terms of hours, minutes, and seconds, then displays the time one second later.

```
Algorithm Time;
Variables hour, minute, second: integer;                    (0.5 point)
Begin
Write ("Enter time in terms of hours, minutes, and seconds");   (0.5 point)
Read (hour, minute, second);
second ← second + 1;
If  (second = 60) then
   second ← 0;
   minute ← minute + 1;
If  (minute = 60) then
    minute ← 0;                                             (3 points)
    hour ← hour + 1;
   If (hour = 24) then
      hour ← 0
   EndIf
 EndIf
EndIf
Write ("Time :", hour, ":", minute, ":", second);
END
```

Exercise n°2 (4 points)

We have the sequence: U0=1, U1= 3, Un=Un-1+2*Un-2 , where n>=2.
Write the algorithm that calculates and displays the term Un, n is a positive integer.

```
Algorithm Sequence,
Variables n, U, A, B, C: integer;                           (0.5 point)
Begin
Repeat
Write ("Enter a positive integer");                         (0.5 point)
Read (n);
Until (n>=0);
If (n = 0) then
   U ← 1;
ElseIf (n = 1) then
   U ← 3;
Else
```

```
   A ← 1;
   B ← 3;
   For i ← 2 to n do
      C ← B + 2 * A;
      A ← B;
      B ← C;
   EndFor
   U ← B ;
EndIf
EndIf
Write ("the term U for",n,"=",U);
END
```

(3 points)

Exercise n°3 **(6 points)**

Consider T an array whose size is N = 100. Write an algorithm that declares this array, requests an integer *n* that indicates the array's real size, and then performs the following operations:

a. Fill the array and display them.
b. Find the length of the longest consecutive repetition.

Example: **Input**: 2 2 0 1 1 1 0 3 4 3 0 5 3/ **Output**: 3 (because 1 repeats 3 times consecutively).

c. Place all zeros at the end of the array while maintaining the order of the elements. From the previous example; **Output** : 2 2 1 1 1 3 4 3 5 3 0 0 0
d. Remove duplicate elements and display the result. **Output** : 2 1 3 4 5 0

```
Algorithm oprerations_on_arrays;
Variables   T: array[1..100] integer;
            n, i, j, k, count, maxLength : integer;
Begin
Write ("Enter real dimension of T");
Repeat
    Read (n);
Until (n>=1 and n<=100)
For i←1 to n do
     Write ("Enter the element n°", i);
     Read (T[i]);
Endfor
For i←1 to n do
    Write ("The element n°", i,"is");
    Write (T[i]);
Endfor
```

(0.5 point)
(0.5 point)
(0.5 point)
(0.5 point)

```
count ← 1
maxLength ← 1
For i ← 2 to n do
   If T[i] = T[i-1] then
      count ← count + 1;
         If (count > maxLength) then
            maxLength ← count;
         EndIf
    Else
      count ← 1;
   EndIf
EndFor
Write ("the length of the longest consecutive repetition is", maxLength);
```

(1.5 point)

```
k ← 1;
For i ← 1 to  n do
   If (T[i] <> 0) then
      T[k] ← T[i];
      k ← k + 1;
   EndIf
EndFor
For i ← k to n do
   T[i] ← 0;
EndFor
```

(1 point)

```
i ← 1; New_size ← n;
While (i ≤ n) do
   j ← i + 1;
   While (j ≤ n) do
      If (T[i] = T[j]) then
         For k ← j to n − 1 do
            T[k] ← T[k + 1];
         EndFor
         New_size← New_size – 1;
      Else
         j ← j + 1;
      EndIf
   EndWhile
   i ← i + 1;
EndWhile
Write ("The new array is ");
For i ← 1 to New_size do
  Write (T[i]);
Endfor\
END
```

(1 point)

(0.5 point)