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

Level: 1st year(Mathematics+CS (MI)) **Date:** 15/05/2025

Module: Algorithmic and Data Structures 2 **Duration**: 1h30m

Exam

Typical solution

Exercice n°01: **(10 points)**

1. **(2 pts)**

```
Function Nbr_Div (n: integer): integer;
Variable i, S: integer;
Begin
   S ←2 ;
   For i ←2 to (n div 2) do
         If (n mod i = 0) then
               S ← S+i ;
         Endif
   EndFor
   Nbr_Div ← S;
End;
```

2. **(2 pts)**

```
Procedure P_Prime (n: integer);
Variable S: integer;
Begin
   S← Nbr_Div (n);
   If (S = 2) then
      write (n, "is a prime number");
   Else
      write (n, "is not a prime number");
   Endif
End;
```

3. **(2 pts)**

```
Function F_Prime (n: integer): boolean;
Variable S: integer;
Begin
   S← Nbr_Div (n);
   If (S = 2) then
      F_Prime ←true;
   Else
      F_Prime ←false;
   Endif
End;
```

4. **(2 pts)**

```
Algorithm Prime;
Variables Bool : boolean, n: integer; // here Bool and n are global variables
//Declaration and definition of subroutines 1, 2, 3.
Function Nbr_Div (n: integer): integer; // here, "n" is a formal parameter.
Variable i, S: integer; // here i and S are local variables
Begin
   S ←2 ;
   For i ←2 to (n div 2) do
         If (n mod i = 0) then
               S ← S+i ;
         Endif
   EndFor
   Nbr_Div ← S;
End;
Procedure P_Prime (n: integer); // here, "n" is a formal parameter.
Variable S: integer; // here  S is local variables
Begin
   S← Nbr_Div (n);
   If (S = 2) then
      write (n, "is a prime number");
```

```
   Else
      write (n, “is not a prime number”);
   Endif
End;
Function F_Prime (n: integer): boolean; // here, “n” is a formal parameter.
Variable S: integer; // here S is local variables
Begin
   S← Nbr_Div (n);
   If (S = 2) then
      F_Prime ←true;
   Else
      F_Prime ←false;
   Endif
End;
BEGIN
   Write ("Enter a number"); Read (n);
   // Procedure call
   P_Prime (n); // here, “n” is an effective parameter.
   // Function call
   Bool ← F_Prime(n); // here, “n” is an effective parameter.
   If (Bool = true) then
      Write (n, “is a prime number”);
   Else
      Write (n, “is not a prime number”);
   End if
END.
```

5. local and global variables, formal and effective parameters (in main algorithm) **(2 pts)**

Exercice n°02: **(10 points)**

```
#include <stdio.h>
void ACQUIRE( int *);
int Fact( int);
void READ_VECTOR (int[], int N);
void CALCULATE_VALUES(int [], int [], int N);
void DISPLAY_TABLE ( int [], int [], int N); (1 pts)
main(){
int X[50]; /* values of X */
int V[50]; /* values of F(X) */
int Nb;
ACQUIRE(&Nb); /* 1 <= Number <= 50 */
READ_VECTOR(X, Nb);
CALCULATE_VALUES(X, V, Nb);
DISPLAY_TABLE(X, V, Nb);
} (2 pts)
void ACQUIRE( int *N)     (1 pts)
{do{
printf ( "Enter an integer between 1 and 50: ");
scanf ("%d",N);
} while ((*N<1)||(*N>50));
}
void READ_VECTOR ( int T[], int N) (1 pts)
{int i;
printf ( "Enter %d real numbers:\n", N);
for (i=0; i<N; i++)
scanf ( "%f", &T[i]);
}
int Fact( int X)          (2 pts)
{  if  (X==0)
        return 1;
   else
        return  X*F(X-1);
}
```

```
void CALCULATE_VALUES( int X[], int V[], int N)
{
int i;
for (i=0; i<N; i++)
   V[i] = F(X[i]);
}
```

(2 pts)

```
void DISPLAY_TABLE(int X[], int V[], int N)
{int i;
printf ( "\nX: ");
for (i=0; i<N; i++)
      printf ( "%d", X[i]);
printf ( "\nFact(X): ");
for (i=0; i<N; i++)
      printf ( "%d", V[i]);
}
```

(1 pts)