University of Oum El Bouaghi
Faculty of Exact Sciences, Natural, and Life Sciences
Department of Mathematics and Computer Science

Level: 1st year(Mathematics+MCS(MI)) **Date:** 16/01/2024

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

Exam

Typical solution

Exercise n°1 (6 points)

Consider the following program:

- Correct the four syntactic errors present in the code. (1 pts)
- Run the program for n=16 and n=27. (2 pts)

n=16 => x=4 n=27 => x=5

- What does this code do? (1 pts)

This program calculates the integer square root of number

- Rewrite the program using a **while** loop. (2 pts)

```
x = n;
i = 0;
while (i < x){
  x = (x + i) / 2;
  i = n / x;
}
```

```
#include <stdio.h>
int main() {
int x, i, n;
printf("Enter an integer: ");
scanf("%d", &n);
x = n;
for (i = 0; i < x; i = n / x)
  x = (x + i) / 2;
printf("%d\n", x);
return 0;
}
```

Exercise n°2 (6 points)

```
Algorithm somme;
var s, x, b :real;
  i, n, f:integer;        (0.5 pts)
begin
write("Give a number :") ;
read(x); (0.5 pts)
write("Give the upper limit");
read(n) ; (0.5 pts)
s←0 ; b←1 ; f←1 ; (0.5 pts)
for i←1 to n-1 do
 b←b*4*x*x;
 f←f*i;
 s←s+b/f;
end for; (3 pts)
write("s=",s); (0.5 pts)
end. (0.5 pts)
```

Exercise n°3 (8 points)

```
Algorithm Negative_Positive_Odd ; (4 points)
Variables n , i, Nboddpos , Nboddneg : integer ;
          T: array[1..50 ] integer; (0.5 pts)
Begin
   Repeat
      Write ("Give the size of  array <=50 ");
      Read(n);
   Until (n>0) and (n≤50); (0.5 pts)
   write ("Enter the elements of  array:");
   Nboddpos ←0 ;
   Nboddneg ←0 ; (0.5 pts)
   For i ← 1 to n do
      Read (T [i]); (0.5 pts)
      If  (T[i] > 0) then (1.5 pts)
            If  (T[i] mod 2= 1) then
               Nboddpos ← Nboddpos + 1 ;
            endif
      else
            If  (T[i] mod 2= 1) then
               Nboddneg ← Nboddneg + 1;
            endif
      endif
   Endfor
Write ("Number of positive odd values: ", Nboddpos );
Write ("Number of negative odd values: ", Nboddneg ); (0.5 pts)
END.
```

First solution **(2 points)**

Algorithm sort _Ascending ;

Variables T: array [1..50] integer;

n, i, j,temp: integer;(0.25 pts)

Begin

Repeat

Write ("Give the size of the array <=50: ");

Read (n);

Until (n>0) and (n≤50); (0.25 pts)

Write (" Enter the elements of array: ");

For i ← 1 to n ***do***

Read (T [i]);

Endfor(0.25 pts)

For i ← 1 to n-1 ***do***

For j ← i+1 to n ***do***

If **(**T[j]< T[i]) ***then***

temp= T[i];

T[i]= T[j];

T[j]= temp;

endif

Endfor

Endfor (1 pts)

*/*Display sorted array*/* ***Write*** ("The sorted array is");

For i ← 1 to n ***do***

Write ("|", T[i], "|");

Endfor(0.25 pts)

END.

Solution by insertion **(2 points)**

Algorithm sort _Ascending ;

Variables T: array [1..50] of integer;

n, i, j, temp: integer; (0.25 pts)

Begin

Repeat

Write ("Give the size of the array <=50: ");

Read (n);

Until (n>0) and (n≤50); (0.25 pts)

Write (" Enter the elements of array: ");

For i ← 1 to n ***do***

Read (T [i]);

(0.25 pts)

Endfor

j ← 2 ;

While (j<n) ***do***

i ←j-1 ;

temp <- T[j];

While (i>0 AND T[i]> temp) ***do***

T[i+1] ← T[i] ;

i ← i-1;

Endwhile

T[i+1] <- temp ;

j ← j+ 1;

Endwhile(1 pts)

*/*Display sorted array*/* ***Write*** ("The sorted array is");

For i ← 1 to n ***do***

Write ("|", T[i], "|");

Endfor(0.25 pts)

END.