Larbi Ben M'hidi-Oum El Bouaghi- University

Faculty of Exact Sciences and Natural and Life Sciences

Mathematics and Computer Science Department

Level: 1st year “Computer science”	**Module:** Algorithmic and Data Structures 2	**Date :** 15/05/2024 **Duration**: 1h30m

Exam n°2

Typical correction

Exercise n°1

```
Algorithm car ;
Type  Car=Record          (1 pt)
                    registration_number: integer;
                    brand: string(20);
                     model: string(20);
                     price: real;
  EndRecord
Variables                    (0.5 pt)
               C : array [1..20] of Car;
               i, pr, ind: integer;
Begin
Write (“Enter the information about the cars”);
For  1←i to 20 do    ( 1.5 pt)
             Read (C[i]. registration_number);
             Read (C[i]. brand);
              Read (C[i]. model);
              Read (C[i]. price);
Endfor
pr← C[1].price ; ind← 1 ; ( 0.5 pt)
```

```
For  2←i to 20 do     ( 2 pts)
  If (C[i].price>pr) then
        pr ← C[i].price ;
         ind← i;
  endif
Endfor
Write (“The most expensive car is the”, ind, “car”);   ( 0.5 pt) // The student displays only
the index of the most expensive car or the details of the most expensive car (both
solutions are correct)
END
```

Exercise n°2

1. **Function Product (V1, V2: [1..20] array of integer): integer** **(1.5 pt)**

```
Variable P : integer ;
Begin
   P ← 0 ;
For  i←1 to N do
   P ← P +(V1[i] * V2[i]);
Endfor
   Product ←  P ;
 End ;
```

2. **Procedure Sum (V1, V2: [1..20] array of integer)** **(1.5 pt)**

```
Variables S1, S2: integer;
Begin
S1=0; S2=0;
For  i←1 to N do
   S1 ← S1 +V1 [i];
   S2 ← S2 +V2 [i];
Endfor
Write (“the sum of the elements of the vector V1 is”,S1);
Write (“the sum of the elements of the vector V2 is”,S2);
End;
```

3. **Algorithm exercise_2;**

Function Product (V1, V2: [1..20] array of integer): integer

```
    Variable P : integer ;
    Begin
      P ← 0 ;
    For  i←1 to N do
      P ← P +V1 [i] * V2 [i];
    Endfor
      Product ←  P ;
     End;
Procedure Sum (V1, V2: [1..20] array of integer);
    Variables S1, S2: integer;
    Begin
    S1=0; S2=0;
    For  i←1 to N do
      S1 ← S1 +V1 [i];
      S2 ← S2 +V2 [i];
    Endfor
    Write (“the sum of the elements of the vector V1 is”,S1);
    Write (“the sum of the elements of the vector V2 is”,S2);
    End;
Variables V1: array [1..20] of integer; V2: array [1..20] of integer; prod, N, i integer;      (0.5
pt)
    Begin
        Repeat     ( 0.5 pt)
            Write (“Enter the real dimension of the vectors”);
            Read(N);
       Until (N>=1 & N<=20)
      Write (“Enter the elements of the first vector V1”);
       For  1←i to N do    ( 0.5 pt)
          Read (V1[i]);
       Endfor
      Write (“Enter the elements of the second vector V”);
       For  1←i to N do      ( 0.5 pt)
```

```
        Read (V2[i]);
      Endfor
    prod ← Product (V1, V2);   (0.5 pt)
    Write (“Scalar product of the two vectors V1 and V2 is”, prod);
    Sum (V1, V2);  (0.5 pt)
    END
```

4. **No, we can’t;** **(0.5 pt)**

   Justification: The variables “S1 & S2”, declared to calculate the sum of the elements of the array, **are local variables** . Indeed, they cannot be used in the main algorithm to compare between S1 and S2 because the space of these local variables is freed when the execution of the procedure is finished.

   Solution: there are three possible solutions **(1.5 pt)**

   1. Use S1 and S2 as parameters and pass them by variable (var S1, S2).
   2. Declare S1 and S2 as global variables.
   3. Use two different function the first one to calculate S1 and the second to calculate S2.

Exercise n°3

```
Algorithm Find_indexes_odd ;
  Variables T : array [1..100] integer ; P, P1:*integer ;  (0.5 pt)
Begin
For P=T to (T+100) do  (1.5 pt)
    Read (*P);
Endfor
Allocate (P1);  (1.5 pt)

*P1←0;
For P=T to (T+50) do
  If ((*P) mod 2 != 0) then                              (2.5 pts)
      Write("index of the odd number is”,P-T);
      *P1←*P1+*P;
  Endif
End
```