Ministry of Higher Education and Scientific Research
HassibaBenbouali University of Chlef
Faculty of Exact Sciences and Computer Science
1st year MI
Module: Algorithmics and data structure 1

Tutorial N° 06 : Records

Exercise 01 :

Create a record named « Student » which is characterized by an identifier, a last name and a first name. You are asked to enter 10 students, arrange them in a table and then display them.

Solution:

```
Algorithm ex1
Struct Student= record
        id: integer;
        lname:c.c;
        fname:c.c;
    end record;
var i:integer;

 ST= array [10]:Student;

Begin
write (" enter the informations of 10 students \n ");
for(i=1 to 10 step 1) do
   write (" enter the informations of the student number \n",i);
   read(ST[i].id,ST[i].lname, ST[i].fname);
end for;
write (" display the information of the students \n");
for(i= 1 to 10 step 1 )do
   write (" the profile of the student ",i , "is\n");
   write(ST[i].id,ST[i].lname, ST[i].fname);
end for;
end
```

Exercise 02 :

We repeat the previous exercise but we add two grades for each student. You are therefore asked to create a new record named « Notes » which this time contains the continuous assessment grade noted « NoteCC » as well as the exam grade which is noted « NoteEx ». Edit the previous « Student » record so that it can relate to the « Notes » one. You are asked to write:

1- An algorithm that allows you to enter the list of students as well as their grades.
2- An algorithm that allows you to display the list of students with their grades.
3- An algorithm that displays the student who had the best exam grade.
4- An algorithm that displays the overall class average.

Solution:

```
Algorithm ex2
Struct Notes= record
        id: integer;
        lname:c.c;
        fname:c.c;
        notecc:real;
        noteex:real;
```

```
        end record;
var i , pos:integer;

 ST= array [10]:Student;
 max,av,sum : real;
Begin
sum←0;
write (" enter the informations of 10 students \n ");
for(i=1 to 10 step 1) do
    write (" enter the informations of the student number \n",i);
    write (" enter the id laste name first name notecc and note ex \n",i);

    read(ST[i].id,ST[i].lname, ST[i].fname, ST[i].notecc, ST[i].noteex);
    sum←sum+( ST[i].notecc+ ST[i].noteex*2)/3;
end for;
write (" display the information of the students \n");
for(i= 1 to 10 step 1 )do
    write (" the profile  and the grades of  the student ",i , "are\n");
    write(ST[i].id,ST[i].lname, ST[i].fname, ST[i].notecc, ST[i].noteex);
end for;
write("calculate the best exam grade \n");
max←St[1].noteex;
pos←1;
for (i=2 to 10 ) do
    if (St[i].noteex >max) then
          max←St[i].noteex;
          pos←i;
   endif;
end for;
write (" the student who had the best grade is \", ST[pos].id, ST[pos].lname, ST[pos].fname,
ST[pos].notecc, ST[pos].noteex);
write("display the average of the class\n");
av←sum/10;
write ("the average of the class is",av);

end
```

Exercise 03 :

Define a « Date » record containing the day, month, and year. Enter two dates (we will assume that they are valid dates) and display whether the first date is before or after the second.

Solution

```
Algorithm ex03
Struct date=record
         day :integer ;
        mon :integer ;
        yea : integer ;
        end record
var d1,d2: date;
begin
write ("enter two correct dates \");
repeat
read(d1.day , d1.mon, d1.yea);
until ((d1.day >0) and  (d1.day≤31) and (d1.mon >0) and  (d1.mon≤12) and (d1.yea >0) and
(d1.yea≤2025))
```

```
repeat
read(d2.day , d2.mon, d2.yea);
until ((d2.day >0) and (d2.day≤30) and (d2.mon >0) and (d2.mon≤12) and (d2.yea >0) and (d2.yea≤2025))
\\comparaison
if (d1.yea>d2.yea) then
 write("the first date is after the second \n");
 else
 if(d1.yea<d2.yea)then
  write("the first date is before the second \n");
  else
   if (d1.mon>d2.mon) then
   write("the first date is after the second \n");
    else
     if(d1.mon<d2.mon)then
      write("the first date is before the second \n");
       else
        if (d1.day>d2.day) then
        write("the first date is after the second \n");
        else
          if(d1.day<d2.day)then
          write("the first date is before the second \n");
          else
          write("the two dates are the same \n");
         endif
       endif
     enif
   endif
  endif
 endif
End
```

Supplementary exercises

Exercise 04 :

Declare the types that allow you to store:

1- A basketball player characterized by his name, date of birth, nationality, and gender.
2- An association of basketball players.
3- A basketball team with its name, its players, as well as the points won and the basketballs scored and conceded in the current season.

Exercise 05 :

We are interested in the management of vehicles in a car fleet. Each vehicle is characterized by a car registration number, a brand, a model, a color, the number of seats, a tax power.

1- Give the adequating record.
2- Break down the car registration number into its elementary components then give the new structure of the record.
3- Write an algorithm that allows you to:
- Store information about a car fleet that includes a maximum of 50 vehicles.
- Establish a statistical array containing the number of vehicles registered per wilaya.

Exercise 06 :

Consider the following types of records:

```
Type   Date = Record
            day, month, year: integer;
       EndRecord;
       Adress = Record
            Number: integer;
            Street: string [50];
            City: string [20];
            Wilaya: string [20];
            PC: integer; /*Postcode*/
       EndRecord;
       Inhabitant = Record
            Last_name, first_name: string [20];
            Date_birth: Date;
            Residence: Address;
       EndRecord;
```

Write an algorithm allowing to:

1- Complete an array TInh with n inhabitants ($n \leq 100$).
2- Display from TInh the addresses of residents born before a given year of birth.
3- Display the names and the birth dates of the inhabitants of the « Ouled Fares » town in the wilaya of « Chlef ».
4- Edit the inhabitants number per wilaya.