

SOLUTION of SW N°1: RECORDS AND FILES

Exercise 1 : The library (Exam ADS-II 2020-2021)

Consider the following records :

```
Type      Date = Record
           day, month, year : integer ;
           EndDate

           Person = Record
           Last_name, first_name: string [30] ;
           Date_birth : Date ;
           Nationality : string ;
           EndPerson

Book = Record
   ISBN : string [12] ;
   title : string [50] ;
   language : string [25];
   Date_edition : Date ;
   pages_nbr : integer ;
   Author : Personne ;
   EndBook
```

Let **T** be an array of at most 1000 books.

1. Display, from T, books written by Algerians in French.
2. Show the names of authors who published a book after the age of 60.
3. Find and view the book with the most pages.

```
Algorithm bibliotheque;
Type      Date = Record
           day, month, year : integer ;
           EndDate

           Person = Record
           Last_name, first_name: string [30] ;
           Date_birth : Date ;
           Nationality : string ;
           EndPerson

           Book = Record
           ISBN : string [12] ;
           title : string [50] ;
           language : string [25];
           Date_edition : Date ;
           pages_nbr : integer ;
           Author : Person ;
           EndBook

Var      T : array[1..1000] of Book ;
         n, i, max, indice: integer;

Begin
  //Fill the array of Records
  Read(n) ;
  For i := 1 to n Do
    Read (T[i].ISBN) ;
    Read (T[i].title) ;
    Read (T[i].language ) ;
    Read (T[i].Date_edition.day) ;
    Read (T[i].Date_edition.month) ;
    Read (T[i].Date_edition.year) ;
    Read (T[i].pages_nbr ) ;
    Read (T[i].Author.Last_name) ;
    Read (T[i].Author.first_name) ;
    Read (T[i].Author.Date_birth.day) ;
    Read (T[i].Author.Date_birth.month) ;
    Read (T[i].Author.Date_birth.year) ;
    Read (T[i].Author.Nationality) ;
  Endfor
```

```

//1. Display, from T, books written by Algerians in French
For i := 1 to n Do
    if (T[i].Author.Nationality = "Algerian" et T[i]. language = "French") then
        Ecrire("Title : ", T[i].Title);
    endif
Endfor

//2. Show the names of authors who published a book after the age of 60.
For i := 1 to n Do
    if (T[i]. Date_edition.year - T[i].Author.Date_birth.year) >= 60) then
        Ecrire("last name : ", T[i].Author.Last_name);
        Ecrire("first name : ", T[i].Author.First_name);
    endif
Endfor

//3. Find and view the book with the most pages.
max ← T[1].pages_nbr;
indice ← 1;
For i := 2 to n Do
    if (T[i].pages_nbr > max) then
        max ← T[i].pages_nbr;
        indice ← i;
    endif
Endfor
Write(' the book with the most pages is ', T[indice].Title,' with ',max,' pages') ;
Fin.

```

Exercise 2 : Residents

Consider the following records :

```

Type Date = Record
    day, month, year: integer ;
EndDate

Address = Record
    Number : integer ;
    Street: string[50] ;
    City, Wilaya: string[25];
EndAddress.

Resident = Record
    Last_name, first_name: string[50];
    Date_naiss: Date ;
    Residence: Address;
EndResidant

```

1. Write a program that allows you to:

- a) Fill an array T with N residents ($N \leq 100$)
- b) Display from T the residents born before a given year « birth_year”
- c) Display the names and dates of birth of the residents of the city of “zemmouri” in the wilaya of “Boumerdès”
- d) Save in a file G of type resident, the residents of the wilaya of Algiers appearing in array T.

```

Algorithme habitants;
Type      Date = Enregistrement
            Jour, mois, année : entier ;
            Fin
            Adresse = Enregistrement
            Numero: entier ;
            Rue : chaîne [50] ;
            Ville, Wilaya: chaîne [25];
            Fin
            Habitant = Enregistrement
            Nom, Prenom : chaîne[50];
            Date_naiss: Date ;
            Residance : Adresse ;
            Fin
Var      T : tableau[1..100] de Habitant ;
            G : Fichier de Habitant ;
            N, i, annee_naiss: entier ;

Début
┌ //1. Remplir le tableau d'enregistrement avec N habitants
├ Lire(N) ;
├ Pour i de 1 à N Faire
├ | Lire (T[i].Nom) ;
├ | Lire (T[i].Prenom) ;
├ | Lire (T[i].Date_naiss.Jour) ;
├ | Lire (T[i].Date_naiss.mois) ;
├ | Lire (T[i].Date_naiss.annee) ;
├ | Lire (T[i].Residance.Numero) ;
├ | Lire (T[i].Residance.Rue) ;
├ | Lire (T[i].Residance.Ville) ;
├ | Lire (T[i].Residance.Wilaya) ;
├ Fpour
├ //2.Afficher à partir de T les habitants nés avant une année donné
├ Lire(annee_naiss) ;
├ Pour i de 1 à N Faire
├ | Si (T[i].Date_naiss.annee < annee_naiss) Alors
├ |   Ecrire("Nom : ", T[i].Nom);
├ |   Ecrire("Prenom: ", T[i].Prenom);
├ |   ...
├ | Fsi
├ Fpour
├ //3.Afficher les noms et les dates de naissances des habitants de la ville de
├ // « zemmouri » de la wilaya de « Boumerdès »
├ Pour i de 1 à N Faire
├ | Si (T[i].Residance.Ville = "zemmouri" et T[i].Residance.Wilaya = "Boumerdès") Alors
├ |   Ecrire("Nom et Prenom: ", T[i].Nom," ",T[i].Prenom);
├ |   Ecrire(T[i].Date_naiss.Jour, "-", T[i].Date_naiss.mois, "-", T[i].Date_naiss.annee);
├ | Fsi
├ Fpour
├ //4. Sauvegarder dans G, les habitants de la wilaya d'Alger
├ ASSIGN(G, "Habitants_Alger") ; /* Assigner le fichier G */
├ Rewrite(G) ; /* Ouvrir le fichier G en écriture pour la 1ère fois */
├ Pour i de 1 à N Faire
├ | Si (T[i].Residance.Wilaya = "Alger") Alors
├ |   Write(G,T[i]) ; /* Ecrire les habitants d'Alger dans le Fichier G */
├ | Fsi
├ Fpour
├ Close(G) ;
└ Fin.

```

Exercise 3 :

Consider the following record :

```
Type Student = Record
    Matricule : integer ;
    Last_name, first_name: string ;
    Score : Real ;
EndStudent.
```

Let **T** be an array of at most 100 students.

1. Write an algorithm to copy all admitted students belonging to **T** into an "ADMIS" file. A student is admitted if his score is greater than or equal to 10.

Solution :

```
Algorithme classe;
Type      Etudiant= Enregistrement
    Matricule: entier ;
    Nom, Prenom : chaîne[50];
    Moyenne : réel ;
Fin

Var      T : Tableau[1..100] de Etudiant ;
    F : Fichier de Habitant ;
    N, i : entier ;

Début
    //1. Remplir le tableau d'enregistrement avec N étudiants
    Lire(N) ;
    Pour i de 1 à N Faire
        Lire (T[i].Matricule) ;
        Lire (T[i].Nom) ;
        Lire (T[i].Prenom) ;
        Lire (T[i].Moyenne) ;
    Fpour

    //1. Copier les étudiants admis appartenant à T dans un fichier ADMIS

    ASSIGN(F, "ADMIS") ;           /* Assigner le fichier F */
    Rewrite(F) ;                   /* Ouvrir le fichier F en écriture pour la 1ère fois */

    Pour i de 1 à N Faire
        Si (T[i].Moyenne >= 10) Alors
            Write(F,T[i]) ;        /* Ecrire les étudiants admis dans le Fichier F */
        Fsi
    Fpour

    Close(F) ;
Fin.
```

Exercise 4 :

Consider the following type :

```
Type Product = Record
    Code : integer ;
    Designation : string[100] ;
    Price : Real ;
EndProduct.
```

Let F be a file of products.

1. Write an algorithm to display the most expensive product in file F.

Solution :

```
Algorithme magasin;
Type      Produit = Enregistrement
    Code : entier ;
    Designation : chaîne[100] ;
    Prix : réel ;

    Fin

Var      F : Fichier de Produit;
    Prod :Produit ;
    Prix_max : Réel ;
    Design_max : Chaîne[100] ;
Début
    ASSIGN(F, "Produits") ;          /* Assigner le fichier F */
    Reset(F) ;                      /* Ouvrir le fichier F en lecture */

    Prix_max ← 0;
    Design_max ← "";

    TQ (Non(EOF(F)) Faire           /* ou : EOF(F) = faux */
    Read(F, Prod) ;                 /* Lire l'élément suivant dans le Fichier F */
    Si (Prod.Prix > Prix_max) Alors
        Prix_max ← Prod.Prix
        Design_max ← Prod.Designation ;
    Fsi
Ftq

    Close(F) ;

    Ecrire("Le produit le plus cher est :", Design_max) ;
Fin.
```