



Module : Programmation et structure de données

Semestre : 02

T.D. N° 04

Les Enregistrements

Exercice n° :01

Définir un type **TEMPS** qui contient les **champs** : **heure, minute, seconde**.

Exercice n° :02

Ecrire une **Procédure** qui réalise la **somme T** de deux durées **T1** et **T2** de type temps.

Exercice n° :03

Ecrire une fonction **TRANSFORM** qui transforme un temps **T** de type **TEMPS** en un entier **S** qui exprime ce temps en **secondes**.

Exemple : pour **T = 3 heures 15 minutes 57 secondes**, **S = 11757 secondes**.

Exercice n° :04

Ecrire une procédure **DECOMPOS** qui décompose un temps **S** exprimé en secondes en un temps **T** de type **TEMPS**.

Exemple : pour **S = 4968 secondes**, **T = 1 heures 22 minutes 48 secondes**.

Exercice n° :05

Etant donnés deux temps **T1** et **T2** de type **TEMPS**, écrire un algorithme qui calcule le temps **T** somme des temps **T1** et **T2** (**T**, **T1** et **T2** sont de type **TEMPS**) en utilisant les (la Fonction et la Procédure) **TRANSFORM** et **DECOMPOS**.

Exercice n° :06

Soit **Date** un type date composé des champs entiers **JJ, MM, AA**.

- Ecrire une Fonction **CompareDates** permettant de comparer deux dates **Date1** et **Date2**.

- Soit **TDate** un tableau de **N** dates ($N \leq 100$).

En utilisant Fonction **CompareDates**, écrire un algorithme permettant de trier ce tableau dans l'ordre croissant des dates.

Exercice n° :07

Ecrire une fonction **Diff-Jours** qui détermine la différence en nombre de jours entre deux dates.

Exercice n° :08

Soit un enregistrement **E** défini par deux informations :

- **T** un tableau d'entiers pouvant contenir au maximum **50** éléments;
- **N** le nombre d'éléments du tableau **T**.

Soit une chaîne de caractères **CH**, écrire une Procédure paramétrée qui retourne un enregistrement de type **E** contenant toutes les positions de la chaîne "**ab**" dans la chaîne **CH**.

Exemple : **CH** = "faabaababbaabrs", Positions : **3 - 6 - 8 - 12** , Nombre d'éléments : **4**

Exercice n° :09

Considérons les types d'enregistrements suivants :

Type Date = Enregistrement

Jour, mois, année : entier ;

FinEnregistrement;

Type Habitant = Enregistrement

Nom, Prenom : chaîne [20] ;

Date_naiss : Date ;

Residence : Adresse ;

FinEnregistrement;

Type Adresse = Enregistrement

Numéro : entier ;

Rue : chaîne [50] ;

Ville : chaîne [20] ;

Wilaya : chaîne [20] ;

CW : entier ; { Code Wilaya }

FinEnregistrement;

Ecrire un algorithme permettant de :

- 1- Remplir un tableau **T** de **N** habitants ($N \leq 100$).
- 2- Afficher à partir de **T** les adresses des habitants nés avant une année de naissance donnée.
- 3- Afficher les noms et les dates de naissance des habitants de la ville de "**Zemmouri**" de la wilaya de "**Boumerdes**".
- 4- Editer le nombre d'habitants par wilaya.

Exercice n° :10

On s'intéresse à la gestion des véhicules d'un parc auto. Chaque véhicule est caractérisé par un matricule, une marque, un modèle, une couleur, le nombre de places, une puissance fiscale.

- 1- Donner l'enregistrement permettant de décrire un véhicule.
- 2- Décomposer le matricule en ses composants élémentaires puis donner la nouvelle structure de L'enregistrement.
- 3- Ecrire un algorithme qui permet de :
 - Stocker les informations d'un parc auto regroupent au max 50 véhicules en utilisant les structures adéquates ;
 - Etablir la liste (**matricule, marque, modèle, puissance**) des véhicules d'une couleur donnée.
 - Etablir un tableau statistique contenant le nombre de véhicules immatriculés par wilaya.

Exercice n° :11

Ecrire un algorithme qui lit deux nombres complexes **NC1** et **NC2** et qui affiche en suite leur somme et leur produit.

On utilisera les formules de calcul suivantes :

$$(a + bi) + (c + di) = (a + c) + (b + d)i$$

$$(a + bi) * (c + di) = (ac - bd) + (ad + bc)i$$



Les Enregistrements

NB : En algorithmique, on ne peut pas avoir une fonction de type enregistrement, donc on utilise une procédure.

Exercice n° :01

Type TEMPS=Enregistrement

H, M, S :Entier ;

FinEnregistrement ;

Exercice n° :02

Procédure SommeT(T1,T2 :TEMPS ; T :TEMPS) ;

Var X :Entier ;

Debut

X←T1.S+T2.S;

T.S←X mod 60;

T.M← X div 60;

X←T.M+T1.M+T2.M;

T.M← X mod 60;

T.H← X div 60+ T1.H+T2.H;

FinProcédure ;

Exercice n° :03

Fonction TRANSFORM(T : TEMPS) : Entier ;

Var TRANSFORM : Entier ;

Debut

TRANSFORM←T.S+60*T.M+3600*T.H ;

FinFonction ;

Exercice n° :04

Procédure DECOMPOS(S :Entier ; T :TEMPS) ;

Debut

T.H←S div 3600 ;

S← S mod 3600 ;

T.M←S div 60 ;

T.S← S mod 60 ;

FinProcédure ;

Exercice n° :05

Algorithme CalculT;

Type TEMPS=**Enregistrement** ;

H, M, S :**Entier** ;

FinEnregistrement ;

Var T1, T2 , T :**TEMPS** ;

S : **Entier** ;

Début (Algorithme Principal)

Ecrire('Donner un Temps T1 : H, M et S') ;

Lire(T1.H,T1.M,T1.S) ;

Ecrire('Donner un Temps T2 : H, M et S') ;

Lire(T2.H,T2.M,T2.S) ;

/ transformer T1 et T2 en secondes, puis additionner */*

S← TRANSFORM(T1) + TRANSFORM(T2) ;

*/*décomposer S en TEMPS T */*

DECOMPOS(S , T) ;

*/*On peut faire aussi DECOMPOS(TRANSFORM(T1) + TRANSFORM(T2) ,T) ;*

Ecrire("La somme est :",T.H," :",T.M," :",T.S) ;

Fin.

Exercice n° :06

Type Date = **Enregistrement**

JJ,MM,AA :**Entier** ;

FinEnregistrement ;

On considère une Fonction pouvant prendre **1** Pour **D1>D2**, **0** Pour **D1=D2** et **-1** Pour **D1<D2**

Fonction CompareDates(D1,D2 :Date) :**entier** ;

Var CompareD :**Entier** ;

Debut

Si (D1.AA>D2.AA) **Alors**

CompareD←1 ;

Sinon

Si (D1.AA<D2.AA) **Alors**

CompareD← -1 ;

Sinon

Si (D1.MM>D2.MM) **Alors**

CompareD←-1 ;

Sinon

Si (D1.MM<D2.MM) **Alors**

CompareD← -1 ;

Sinon

Si (D1.JJ>D2.JJ) **Alors**

CompareD←-1 ;

Sinon

```

    Si (D1.JJ<D2.JJ) Alors
        CompareD← -1 ;
    Sinon CompareD← 0 ;
    FinSi ;
    FinSi ;
    FinSi ;
    FinSi ;
FinFonction ;

/* l'Algorithme trier un tableau de Type Date */

Algorithme TrierDate ;

Type Date = Enregistrement
    JJ,MM,AA :Entier ;
FinEnregistrement ;

Var TDate :Tableau[1..100] de type Date ;
    VP :Date ;
    I,J :entier ;
Fonction CompareDates(D1,D2 :Date) :entier ;
-----
Debut (Algorithme Principal)

Repete
    Ecrire("Donner la valeur de N") ;
    Lire(N) ;
Jusqu'à (N>0 et N≤100) ;

/*lecture du tableau des dates */

Pour I de 1 à N Faire
    Lire(TDate[I].JJ, TDate[I].MM, TDate[I].AA);
FinPour;

/*Trier un Tableau de Type Date */

Pour I de 1 à N-1 Faire
    Pour J de I+1 à N Faire
        Si ( CompareDates( TDate[I], TDate[J])=1 ) Alors
            VP← TDate[I];
            TDate[I]←TDate[J];
            TDate[J]←VP ;
        FinSi;
    FinPour;
FinPour;

/*affichage des dates triées */
Pour I←1 à N Faire
    Ecrire(TDate[I].JJ , TDate[I].MM , TDate[I].AA);
FinPour;
Fin.

Exercice n° :07
Exercice n° :08

```

Exercice n° :09

Algorithme Habitant ;

Type Date = **Enregistrement**

Jour, mois, année : **Entier** ;

FinEnregistrement;

Type Adresse = **Enregistrement**

Numéro : **Entier** ;

Rue : chaîne [50] ;

Ville : chaîne [20] ;

Wilaya : chaîne [20] ;

CW : **Entier** ; { Code Wilaya }

FinEnregistrement;

Type Habitant = **Enregistrement**

Nom, Prenom : chaîne [20] ;

Date_naiss : Date ;

Residence : Adresse ;

FinEnregistrement;

Var T :Tableau[1..100] de Type **Habitant** ;

TCW :Tableau[1..48] de type **Entier** ;

I, N, A, CODE : **Entier** ;

Debut (Algorithme Principal)

Repeter

Ecrire("Donner N Habitants") ;

Lire(N) ;

Jusqu'à (N>0 et N≤100) ;

*/*lecture du tableau des Habitants */*

Pour I de 1 à N **Faire**

Lire(T[I].Nom, T[I].Prenom) ;

Lire(T[I].Date_naiss.Jour, T[I].Date_naiss.mois, T[I].Date_naiss.année) ;

Lire(T[I].Residence.Numéro, T[I].Residence.Rue, T[I].Residence.Ville, T[I].Residence.Wilaya, T[I].Residence.CW) ;

FinPour ;

*/*affichage des adresses des habitants nés avant une année A */*

Ecrire("Donner une Année A ") ; **Lire**(A) ;

Pour I de 1 à N **Faire**

Si(T[I].Date_naiss.année = A) **Alors**

Ecrire(T[I].Residence.Numéro, T[I].Residence.Rue, T[I].Residence.Ville, T[I].Residence.Wilaya, T[I].Residence.CW) ;

FinSi ;

FinPour ;

/* les noms et les dates de naissance des habitants de la ville de "Zemmouri" de la wilaya de "Boumerdes" */

Pour I de 1 à N **Faire**

Si(T[I].Residence.Ville ="Zemmouri" **ET** T[I].Residence.Wilaya="Boumerdes") **Alors**

Ecrire (T[I].Nom, T[I].Prenom) ;

Ecrire (T[I].Date_naiss.Jour, T[I].Date_naiss.mois, T[I].Date_naiss.année) ;

FinSi ;

FinPour ;

/*Nombre d'habitants par wilaya */

/* initialisé TCW à 0 */

Pour I←1 à 48 **Faire**

TCW[I]←0 ;

FinPour ;

Pour I←1 à N **Faire**

CODE← T[I].Residence.CW; TCW[CODE]← TCW[CODE]+1;

FinPour ;

/*affichage Nombre d'habitants par wilaya */

Pour I de 1 à 48 **Faire**

Ecrire("Nombre d'habitants ", TCW[I] , " Par Wilaya ", I) ;

FinPour ;

Fin.

Exercice n° :10

Exercice n° :11

Algorithme NB_Complexe ;

Type NComplexe=**Enregistrement** ;

a,b :**Reel** ;

FinEnregistrement ;

Procédure SommeZ(Z1,Z2 : NComplexe; Z : NComplexe) ;

Debut

Z.a←Z1.a+Z2.a ;

Z.b←Z1.b+Z2.b ;

FinProcédure;

Procédure DiffZ(Z1,Z2 : NComplexe; Z : NComplexe) ;

Debut

Z.a←Z1.a-Z2.a ;

Z.b←Z1.b-Z2.b ;

FinProcédure;

Procédure ProdZ(Z1,Z2 : NComplexe; Z : NComplexe) ;

Debut

Z.a←Z1.a*Z2.a- Z1.b*Z2.b ;

Z.b← Z1.a*Z2.b+ Z1.b*Z2.a ;

FinProcédure;

Var NC1, NC2, SOMC1C2, PRODC1C2, DIFFC1C2 : **NComplexe** ;

Debut (Algorithme Principal)

Ecrire ("Donner Deux Nombres Complexe NC1 et NC2") ;

Lire(NC1.a, NC1.b) ;

Lire(NC2.a, NC2.b) ;

SommeZ(NC1, NC2, SOMC1C2) ;

DiffZ(NC1, NC2, DIFFC1C2) ;

ProdZ(NC1, NC2, PRODC1C2) ;

Ecrire(SOMC1C2.a, SOMC1C2.b) ;

Ecrire(DIFFC1C2 .a, DIFFC1C2 .b) ;

Ecrire(PRODC1C2.a, PRODC1C2.b) ;

Fin.