

Travaux Dirigés 2

Enregistrements (Structures)

Rappel de cours:

- On déclare un *type* enregistrement comme suit :

```
Type  nomTypeEnreg = enregistrement  
                               liste de champs ;  
                               fin ;
```
- Pour en instancier une variable, on utilise :

```
Var  varEnreg : nomTypeEnreg;
```

Exemple : On veut déclarer deux variables d'une structure que nous appelons tPersonne contenant deux champs : nom, de type tableau de caractères, et age, de type entier :

```
Type  tPersonne=enregistrement  
                               nom : tableau (20) caractere ;  
                               age : entier;  
                               fin ;  
Var   pers1,pers2 :tPersonne;
```

- On peut accéder à un champ d'une variable enregistrement par l'opérateur '.' comme suit :
pers1.age
- Une telle variable peut être manipulée comme n'importe quelle variable du type de champ associé.
- Manipuler une variable enregistrement de façon globale n'est permis que pour l'affectation et la transmission de paramètres.

Exercice 1 :

Un nombre complexe Z est défini par ses parties réelle a et imaginaire b.

1. Ecrire un algorithme qui lit deux nombres complexes Z1 et Z2 et qui affiche ensuite leur somme et leur produit.

- $(a+bi) + (c+id) = (a+b) + (b+d)i$
- $(a+bi) * (c+id) = (ac-bd) + (ad+bc)i$

L'algorithme doit utiliser deux actions paramétrée saisi(), affiche(), addition() et produit() qui permettent respectivement de saisir, afficher, additionner et multiplier des nombres complexes.

2. Traduire votre algorithme en C.

Exercice 2 :

Ecrire un algorithme qui permet de gérer un ensemble d'étudiants (≤ 100). Chaque étudiant est décrit par son identifiant, son nom, son prénom, son âge et sa moyenne.

Ecrire un algorithme qui permet de :

1. Saisir les informations concernant des étudiants.
2. Afficher les informations saisies et le résultat concernant chaque étudiant sachant que le résultat est soit ADM ou AJN (selon que la moyenne est < 10 ou non).
3. Afficher la moyenne générale des étudiants.

Exercice 3 :

1. Définir un type enregistrement TEMPS qui contient les champs heure, minute, seconde.
2. Ecrire une action paramétrée qui réalise la somme T de deux durées T1 et T2 de type TEMPS.
3. Ecrire une action paramétrée transform() qui transforme un temps T de type TEMPS en un entier S qui exprime ce temps en secondes.

Exemple : pour T = 2 heures 10 minutes 37 secondes, S = 7837 secondes.

Ecrire une procédure decompos() qui décompose un temps S exprimé en secondes en un temps T de type TEMPS.

Exemple : pour S = 7837 secondes, T = 2 heures 10 minutes 37 secondes.

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 actions transform() et decompos().

Exercice 4 :

Écrire un programme qui introduit deux temps en (heure/minute/seconde) et calcule la différence entre eux. Le programme utilisera les actions paramétrées suivantes :

- struct TEMPS lecture() // Lecture de temps en HH:MM:SS
- void affiche(struct TEMPS t) // Affichage de temps en HH:MM:SS
- struct TEMPS diff(struct TEMPS start, struct TEMPS end) // calcule la différence entre les temps start et end.
- struct TEMPS decompos(int s) // Convertit un nombre de secondes en HH:MM:SS

L'affichage doit être comme suit :

```
Donnez le premier temps
Donnez les heures :4
Donnez les minutes :51
Donnez les secondes :20
4 : 51 : 20
Donnez le deuxieme temps
Donnez les heures :10
Donnez les minutes :3
Donnez les secondes :44
10 : 3 : 44
Difference entre les deux temps 5 : 12 : 24
Difference en secondes : 18744
```

Exercice 5 :

Considérons les enregistrements suivants :

```
Type
Date = Enregistrement
    Jour, mois, annee : Entier ;
Fin;
Adresse = Enregistrement
    Numero : entier ;
    Rue : chaine [50] ;
    Ville : chaine [20] ;
    Wilaya : chaine [20] ;
```

```

Fin;
Enseignant = Enregistrement
                Nom, prenom           : chaine [20] ;
                Date_naiss            : date ;
                Residence              : Adresse ;
                Grade                  : chaine [10] ;
                Annee_de_recrutement   : entier ;
Fin;

```

Ecrire un algorithme qui permet de :

1. Remplir un vecteur T de N enseignants ($N \leq 100$).
2. Afficher à partir de T les noms et adresses des enseignants nés avant une date donnée.
3. Afficher les noms et les prénoms des enseignants résidents de la ville de 'Boudouaou' de la wilaya de 'Boumerdes'.
4. Afficher le nombre d'enseignants par année de recrutement à partir d'une année donnée.

Exercice 6 :

On souhaite gérer une base de données d'inscriptions pour l'organisation d'un congrès qui dure une journée. Les organisateurs proposent aux participants de s'inscrire pour des repas, ainsi que pour l'hébergement en hôtel. Un participant peut s'inscrire indépendamment aux 2 repas proposés : déjeuner (1500 DA) et/ou dîner (3500 DA) ou aucun. Il n'est pas obligé de prendre un hôtel. S'il en prend un, il peut choisir parmi 2 types d'hôtels différents : 2 étoiles (7500 DA) ou 3 étoiles (10000 DA).

Un participant peut venir accompagné de son conjoint. Dans ce cas, la réservation d'hôtel est identique mais lorsqu'un repas est sélectionné alors il faut en compter 2.

1. Créer un enregistrement `Participant` qui inclut son nom, son prénom ainsi que toutes les autres informations nécessaires à son inscription selon les critères définis ci-dessus.

On privilégiera un enregistrement contenant un nombre minimal de champs. L'enregistrement contient donc les champs suivants :

- Nom et prenom
- Dejeuner : Faux = pas de Déjeuner, Vrai = avec Déjeuner
- Diner : Faux = pas de Diner, Vrai = avec Diner
- Hotel : 1 = pas d'Hôtel, 2 = 2 étoiles, 3 = 3 étoiles
- Seul : Faux = avec conjoint, Vrai = seul

2. Créer une variable `TabPart` qui est un tableau de 100 enregistrements de type `Participant` (Donnez la déclaration de cette variable).

3. Créer une action paramétrée `liste2Etoiles()` qui reçoit un tableau `TabPart` de type `participant` et le nombre de participants pour afficher les nom et prénoms des personnes qui ont choisi de réserver un hôtel 2 étoiles.

4. Créer une action paramétrée `NbDej()` qui accepte comme argument un tableau `TabPart` de type `Participant` et le nombre de participants, puis retourne le nombre de déjeuners à prévoir.

5. Créer une action paramétrée `Montant()` qui calcule, pour un participant donné en argument, le montant de sa facture.

Exercices supplémentaires

Examen 2014 - 2015

Un pharmacien veut gérer les opérations de vente et d'achat de N médicaments différents. Pour cela, il associe à chaque médicament les informations : `NomMédicament` de 20 caractères, `NbBoites` de type entier, `PrixUnitaire` de type réel, `DatePeremption` de type date et `Fournisseur` de 20 caractères. Les dates étant définies à l'aide des informations : `annee`, `mois`, `jour` qui sont de type entier.

1. Proposer les structures `tDate` et `tMédicament` pour contenir de telles informations.
2. Ecrire une fonction pour remplir un tableau `TabMédicament` de structures `tMédicament`.
3. Ecrire une fonction qui affiche tous les médicaments livrés par un fournisseur donné.
4. Ecrire une fonction qui retourne le prix total des médicaments périmés à la date : 04 septembre 2015.
5. Ecrire une fonction qui permet au pharmacien de mettre à jour le tableau `TabMédicament` suite à une opération de vente ou à une opération d'achat d'un médicament. On distinguera ces deux opérations à l'aide d'un paramètre en entrée, de type caractère, qui vaut 'v' pour la vente et 'a' pour l'achat.

Remarque : On suppose que chaque médicament est livré par un seul fournisseur.

Rattrapage 2019 - 2020

La Fédération Algérienne de Football (FAF) veut automatiser la gestion des matchs du championnat. Chaque match est représenté par les noms des deux équipes qui l'ont disputé et le nombre de buts marqués par chaque équipe (voir exemple d'exécution). Pour simplifier, on supposera que le nombre de matchs n du championnat ne dépasse pas 50.

1. Proposer une déclaration d'un type enregistrement (ou structure C) `tMatch` qui permet de représenter les données d'un match du championnat.
2. Écrire une action paramétrée `fillvMatch()` qui permet de remplir les données de n matchs ($n \leq 50$) dans un tableau `vMatch` d'enregistrements `tMatch`.
3. Écrire une action paramétrée `bestStrikeTeam()` qui détermine l'équipe ayant la meilleure attaque (i.e ayant le plus grand nombre global de buts marqués) dans le championnat.

Exemple d'exécution pour la question 2 (ici l'affichage des matchs et en plus, il n'est pas demandé) :

```
Donnez le nombre de matchs :3
***** Saisie match Numero 1 *****
Donnez nom equipe 1 :ESS
Donnez nom equipe 2 :CSC
Donnez nombre but(s) ESS :3
Donnez nombre but(s) CSC :2
***** Saisie match Numero 2 *****
Donnez nom equipe 1 :MOB
Donnez nom equipe 2 :JSK
Donnez nombre but(s) MOB :2
Donnez nombre but(s) JSK :2
***** Saisie match Numero 3 *****
Donnez nom equipe 1 :JSMB
Donnez nom equipe 2 :MOB
Donnez nombre but(s) JSMB :1
Donnez nombre but(s) MOB :3
```

Les matchs que vous avez introduits sont: !!!(cet affichage est en plus)!!!

```
***** Match 1 *****
ESS 3 - CSC 2
***** Match 2 *****
MOB 2 - JSK 2
***** Match 3 *****
JSMB 1 - MOB 3
```