

Module PSD (ALGO2) (LMD M/I)

Série N° 4: Les enregistrements

Exercice 1 :

Soit un type **TEMPS** contenant le résultat obtenu par un athlète dans une course de marathon.

- 1- Ecrire une action paramétrée (AP) **TRANSFORM** qui transforme un temps T de type **TEMPS** en un entier S exprimant ce temps en secondes.
- 2- Ecrire une AP **DECOMPOS** qui décompose un temps S exprimé en secondes en un temps T de type **TEMPS**.
- 3- Soient N participants dans cette course ($N \leq 100$). Chaque athlète est défini par un **Numéro**, un **Nom** et un **Résultat** de type **TEMPS**. Ecrire un algorithme permettant de construire un vecteur contenant les informations des différents athlètes, puis affiche la liste des athlètes (avec résultat) sélectionnés ayant obtenu un résultat inférieur ou égale à la moyenne de tous les athlètes.

Exercice 2 :

Un nombre complexe Z est entièrement défini par ses parties réelle **a** et imaginaire **b** ($Z = a + bi$).

- 1- Donner la déclaration d'un nombre complexe,
- 2- Ecrire les APs: **ReelZ**, **ImagZ** et **MODule** donnant les attributs d'un nombre complexe respectivement: la partie réelle, la partie imaginaire et le module),
- 3- Ecrire les actions paramétrées : **SommeZ**, **DiffZ** et **ProdZ** nécessaires à l'arithmétique sur les complexes, respectivement pour l'addition, la soustraction et la multiplication,
- 4- Ecrire une AP **ConjZ** qui calcule le conjugué d'un nombre complexe.
- 5- Ecrire une AP **EgaleZ** qui teste l'égalité de deux nombres complexes.
- 6- Ecrire une AP **EcrireZ** qui permet d'afficher un nombre complexe.

Soit TC un tableau de N nombres complexes ($N \leq 100$). En utilisant les actions paramétrées précédentes, écrire un algorithme qui :

- Affiche l'élément de **TC** ayant le plus petit module. Puis vérifie l'existence de son conjugué dans **TC**.
- Calcule la somme **Zs** et le produit **Zp** des éléments non nuls du tableau **TC**.
- Calcule et affiche le nombre **Zs-Zp** et son module.

Exercice 3 :

Soit TDate un type date composé des champs entiers JJ, MM, AA. Ecrire les APs suivantes :

- **CompareD** : compare deux dates D1 et D2 (-1, 0, 1 pour inférieur, égale, et supérieur).
- **BIS** : vérifie si une année est bissextile.
- **Valide** : vérifie la validité d'une date.
- **Diffjour** : calcule la différence en jours entre deux dates.

Soit TD un tableau de N dates ($N \leq 100$). Ecrire un algorithme permettant de :

- Trier ce tableau dans l'ordre croissant des dates.
- Calculer la différence minimale en nombre de jour entre ces dates.

Exercice 4 :

Soit un enregistrement E défini par deux informations :

- T un tableau d'entiers pouvant contenir au maximum 100 éléments;
- N le nombre d'éléments du tableau T.

Soit une chaîne de caractères M, écrire une action paramétrée qui retourne un enregistrement de type E contenant toutes les positions de la chaîne 'ab' dans la chaîne M.

Exemple : M = 'faabaababbaabrs'

Résultat :

3	6	8	12
4			

Exercice 5 :

Considérons les types d'enregistrements suivants :

Type TDate = Enregistrement

 Jour, mois, année : entier ;

Fin;

TAdresse = Enregistrement

 Numéro : entier ;

 Rue : chaîne [50] ;

 Ville : chaîne [20] ;

 Wilaya : chaîne [20] ;

 Cw : entier ; // Code Wilaya

Fin;

THabitant = Enregistrement

 Nom, prenom : chaîne [20] ;

 Date_naiss : TDate ;

 Residence : TAdresse ;

Fin;

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 **AN** donnée.
- 3- Afficher les noms et les dates de naissance des habitants de la ville **V** d'une wilaya **W**.
- 4- Afficher le nombre d'habitants par wilaya.

Exercice 6 :

Considérons les types enregistrement suivant :

Type

TModule = Enregistrement Nom : chaîne[10] ; Note : réel ; Coef : entier ; Fin ;

TEtudiant = Enregistrement

 Matricule : entier ; Nom, Prenom : chaîne [20] ;

 M1, M2, M3 : TModule ; Moyenne : réel ;

Fin;

Soit T un tableau d'au plus 100 étudiants.

Ecrire un algorithme permettant de remplir le tableau T, calculer la moyenne de chaque étudiant puis recopier tous les étudiants admis dans un tableau ADMIS de type TEtudiant. Un étudiant est admis si sa moyenne est supérieure ou égale 10.