Faculté d'Électronique et Informatique USTHB, 2017/2018
Département Informatique
Domaine MI, Module ALGO2 (PSD)

TD1 : Enregistrements

Exercice 1 :

1. Définir un type **RESULTAT** décrivant les résultats obtenus lors d'une course de marathon.
2. Ecrire une fonction **TRANSFORMER** qui transforme un résultat **TR** de type **RESULTAT** en un entier S qui exprime ce résultat en secondes.
   Exemple : pour TR=2 h 10 mn 37 sec, S=7837 secondes.
3. Ecrire une procédure **DECOMPOSER** qui décompose un résultat **SR** exprimé en secondes en un autre résultat **TR** de type **RESULTAT**.
4. Etant donnés N athlètes (résultats), en utilisant les actions paramétrées **TRANSFORMER** et **DECOMPOSER,** écrire un algorithme qui construit un vecteur **Résultats** de N athlètes (N≤100) puis, affiche les athlètes (numéros) ayant obtenu un temps au dessous de la moyenne de tous les athlètes.

Exercice 2 :

Soit le type enregistrement suivant :
Etudiant = **Enregistrement** Matricule : chaîne[12]; Nom, Prenom: chaine[20]; Moyenne : réel; **Fin;**
Soit **T** un tableau d'au plus 100 étudiants. Ecrire un algorithme permettant de recopier tous les étudiants admis appartenant à **T** dans un tableau **ADMIS** de type étudiant. Un étudiant est admis si sa moyenne est supérieure ou égale à 10.

Exercice 3 :

Soit **Tdate** un type date composé des champs entiers JJ, MM, AA.
Ecrire une AP **CompareD** permettant de comparer deux dates D1 et D2.
Etant donné un vecteur **Dates** contenant N dates (N≤100). En utilisant l'AP **CompareD**, écrire un algorithme permettant de trier ce vecteur dans l'ordre décroissant des dates.

Exercice 4 :

Ecrire une fonction qui vérifie la validité de deux dates et affiche la différence entre elles exprimée en nombre de jours.

Exercice 5 :

Soient les types suivants :
Date =**Enregistrement** jour:01..31; mois:01..12; année:1970..2018 **Fin**;
Personne =**Enregistrement** nom_personne :chaîne[50]; adresse:chaîne [50]; date_naiss:date **Fin;**
Véhicule =**Enregistrement** num_immatriculation:chaîne[10]; marque:chaîne[30]; modèle:chaîne[30]; propriétaire : chaîne [50] **Fin**;// propriétaire et nom_personne ce sont des synonymes
Etant donnés deux vecteurs **PARCS** et **PERS** contenant respectivement, N véhicules et M personnes (N<100) et (M<20), écrire une action paramétrée permettant l'affichage :

- Des véhicules dont le nom du propriétaire est *'«Mohammed'»*?
- Des véhicules immatriculés à alger?
- Des propriétaires (nom, date de naissance) nées avant une date donnée **DN**.
- Les propriétaires (nom, adresse) des véhicules de marque « **Daewoo** »?
- Tous les propriétaires organisés par marques (marque, propriétaire(s)).

Exercice 6 :

Proposer une structure de données pour stocker les coordonnées d'une ville sur le globe terrestre (admettant que la ville est assimilée à un point et la terre à un plan orthogonal XOY) :

Ecrire une fonction **DISTANCE** qui calcule la distance de Manhattan entre 2 villes. Rappelons que la distance de Manhattan entre deux points A(X, Y) et B(X, Y) est calculée comme suit :
distance_manhattan = $| X_A - X_B| + | Y_A - Y_B |$.
Etant donné un vecteur **Trajectoire** contant N villes (n<10) décrivant une trajectoire. En utilisant la fonction **DISTANCE**, écrire une fonction **LONG_TRAJECOIRE** qui calcule la longueur de cette trajectoire (reliant la première et la dernière ville).

Exercice 7 :

Un nombre complexe z est défini par ses deux parties : réelle a et imaginaire b, tel que $z = a + ib$

1- Proposer une structure de données pour représenter un nombre complexe,
2- Ecrire les fonctions : **ReelZ**, **ImagZ** et **MODule** qui donnent respectivement : la partie réelle, la partie imaginaire et le module d'un nombre complexe ***z.***
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 procédure **ConjZ** qui calcule le conjugué d'un nombre complexe z.
5- Ecrire une fonction **EgaleZ** qui vérifie l'égalité de deux nombres complexes ***z1 et z2***.
6- Ecrire une procédure **EcrireZ** qui permet d'afficher un nombre complexe z.

Soit **TC** un tableau de N nombres complexes (N<=100). En utilisant les actions paramétrés précédentes, écrire un algorithme qui :

- Affiche l'élément de **TC** ayant le plus grand 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 la différence entre **Zs** et **Zp** si elle est imaginaire pur.

On rappelle les formules de calcul suivantes :

- La somme : $(a + bi) + (c + di) = (a + c) + (b + d)i$
- Le produit : $(a + bi) * (c + di) = (ac – bd) + (ad + bc)i$
- Le conjugué : $\overline{a + bi} = a – bi$
- Le module : $|a + bi| = \sqrt{a^2 + b^2}$

Exercice 8 :

Soit **E** un enregistrement 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 **RECHERCHER** qui retourne un enregistrement de type E contenant toutes les positions de la chaîne "ab" dans la chaîne M.

En utilisant l'enregistrement l'action paramétrée **Rechercher,** écrire un algorithme permettant de supprimer toutes les occurrences de la chaîne "ab".

Exemple : M = "faabaababbaabrs"

Résultat :

3	6	8	12
4			

M = "faars"