

EXAMEN DU SEMESTRE 2

ALGORITHMIQUE ET STRUCTURES DE DONNEES II

Exercice 1 : Moyenne de la Classe

Soit une matrice $M(n,10)$ une matrice qui comporte les notes obtenues par les n élèves d'une classe dans 10 matières étudiés, avec $n \leq 200$.

1. Ecrire un Algorithme qui permet de :
 - a. Lire les notes des élèves
 - b. Ecrire une fonction *moy_élève(i)* qui calcule et retourne la moyenne de l'élève n° i
 - c. Afficher le numéro de l'élève qui possède la meilleur moyenne en utilisant la fonction *moy_élève*
 - d. Calculer et afficher la moyenne de la classe, en utilisant –aussi- sur la fonction *moy_élève*

Exercice 2 : Réarranger le tableau

1. Ecrire un algorithme qui permet de réarranger (réorganiser) les éléments d'un tableau comme suit :
Le dernier élément devient le 1^{er}, le 1^{er} devient le 2^{ème}, l'avant dernier devient le 3^{ème}

1	2	3	4	5	6	7
---	---	---	---	---	---	---

7	1	6	2	5	3	4
---	---	---	---	---	---	---

Exercice 3 : Horaires de Trains

Considérons les enregistrements suivants :

```
Type   Time = Enregistrement
        heure, minute, seconde : entier ;
Fin

Personne = Enregistrement
        nom, prenom: chaine [30] ;
        Date_naiss : Date ;
        Adresse : chaine [80] ;
        Ville : chaine [30] ;
Fin
```

```
Horaire = Enregistrement
        Num_train : chaine [30] ;
        Ville_depart : chaine [30] ;
        Ville_arrive : chaine [30] ;
        Temp_depart : Time;
        Temp_arrive: Time ;
        Prix : Réel;
        Conducteur : Personne ;
Fin
```

L'enregistrement « Horaire » représente un horaire de train, la ville de départ et d'arrivé, l'heure de départ et d'arrivé ainsi que le conducteur et le prix de voyage.

Soit **Trains** un tableau d'au plus 100 horaires. À partir de ce tableau :

1. Trouver et afficher le voyage qui prend le plus de temps parmi les horaires de « **Trains** »
2. Affiche les conducteurs qui démarre de leurs propres villes,
3. Sauvegarder dans un fichier, les horaires démarrant ou arrivant à la ville de « boumedfaa »

Exercice 4 : Déplacer le dernier élément vers la tête de la liste

Soit **L** une liste linéaire chaînée de valeurs entières *ordonnées*.

1. Ecrire une procédure **RemoveDuplicate(L)** qui permet de supprimer les éléments avec les valeurs en double de la liste **L**.

Exemple : Si **L** : 2→2→2→3→5→5→6→7 alors **RemoveDuplicate(L)** retourne **L** : 2→3→5→6→7

Bon Courage