

## ***Les Enregistrements***

### ***Exercice 1 :***

Proposer une structure d'enregistrement qui permet de représenter le nom d'un étudiant et la liste de ses notes obtenues pendant les contrôles continus.

Le nombre de notes disponibles peut varier d'un étudiant à un autre d'où l'utilité du champ Nb qui donne le nombre de notes disponibles pour l'étudiant correspondant (au max 10 notes).

1. Écrire un algorithme qui permet de :
  - a. saisir un étudiant avec les notes obtenues
  - b. Calculer la moyenne des contrôles et l'enregistrer dans le champ Moy
  - c. Afficher la moyenne.
2. On désire faire le même travail pour 100 étudiants au plus. Écrire l'algorithme qui permet de :
  1. Lire les données concernant les N étudiants.
  2. Calculer toutes les moyennes
  3. Afficher les noms des étudiants ayant une moyenne  $\geq 10$
  4. Trier les étudiants par ordre décroissant sur la moyenne
  5. Afficher les noms des 3 majors seulement.

Soit la structure d'enregistrement proposée qui permet de représenter les résultats d'un étudiant.

1. Ajouter à cette structure le champ **Adresse** qui est composé du Numéro du bâtiment, le nom de la cité, le nom de la commune et le nom de la wilaya
2. Écrire une fonction qui permet d'afficher les moyennes de tous les étudiants qui habite à la wilaya de Tindouf.
3. Écrire un algorithme qui affiche tous les étudiants qui habitent dans la même commune que l'étudiant ayant la plus basse moyenne.

### ***Exercice 2 :***

Pour calculer les statistiques d'utilisation du vocabulaire d'un texte, on a besoin d'une structure qui enregistre tous les mots du texte et le nombre d'occurrences de chaque mot.

1. Proposer une structure d'enregistrement appropriée.
2. Donner la fonction qui permet de calculer le nombre d'occurrences de chaque mot du texte.

Donner le thème du texte en affichant les mots les plus redondants dans le texte.

## *Les Pointeurs*

**Exercice 1 :** Copiez le tableau suivant et complétez-le pour chaque instruction du programme ci-dessus.

```
main()
{ int A = 1;
  int B = 2;
  int C = 3;
  int *P1, *P2;
  P1=&A;
  P2=&C;
  *P1=(*P2)++;
  P1=P2;
  P2=&B;
  *P1-=*P2;
  ++*P2;
  *P1*=*P2;
  A=++*P2**P1;
  P1=&A;
  *P2=*P1/=*P2;
  return 0; }
```

	<u>A</u>	<u>B</u>	<u>C</u>	<u>P1</u>	<u>P2</u>
Init.	1	2	3	/	/
P1=&A	1	2	3	&A	/
P2=&C					
*P1=(*P2)++					
P1=P2					
P2=&B					
*P1-=*P2					
++*P2					
*P1*=*P2					
A=++*P2**P1					
P1=&A					
*P2=*P1/=*P2					

**Exercice2:** Soit P un pointeur qui 'pointe' sur un tableau A:

```
int A[] = {12, 23, 34, 45, 56, 67, 78, 89, 90};  
int *P;  
P = A;
```

Quelles valeurs ou adresses fournissent ces expressions:

- a)        \*P+2
- b)        \*(P+2)
- c)        &P+1
- d)        &A[4]-3
- e)        A+3
- f)        &A[7]-P
- g)        P+(\*P-10)
- h)        \*(P+\*(P+8)-A[7])

Q: - Expliquer les différences entre (\*p)++ ; \*p++ ; \*(p++)

Q: - Ecrire une fonction d'échange de valeurs de deux variables réelles données en argument

### Exercice 3:

1- Ecrire une fonction de saisie des éléments d'un tableau de réels. Le nombre d'éléments du tableau sera donnée en argument de la fonction qui retournera l'adresse du tableau. Ce tableau devra être allouée dynamiquement. Une gestion des erreurs permettra de renvoyer le pointeur NULL en cas d'erreur d'allocation mémoire.

2- Inverser un tableau de réels en utilisant les pointeurs On utilise les deux notations pointeur et sans pointeur pour passer l'adresse du tableau en argument de la fonction.

3- Ecrire la fonction qui affiche puis libère la mémoire d'un tableau allouée dynamiquement

## *Les Fichiers*

```
- Declaration : FILE *F;
- Ouverture en écriture: F= fopen ( " fichier.txt", "w");
- Ouverture en lecture : F= fopen ( " fichier.txt", "r");
- Parcourir un fichier : while ( !feof (f)) /* feof = f end of the file */
- Lire à partir d'un fichier : fscanf (F, "%s", &X);
- Ecrire dans un fichier : fprintf( F,"%s\n", X) ;
- Fermer un fichier pour mettre la tete au début de fichier : fclose(F) ;
```

### **Exercice 1:**

1. Écrire un algorithme qui crée le fichier MOTS.TXT contenant une série de 50 mots au maximum (longueur maximale d'un mot: 20 caractères). La saisie des mots se terminera à l'introduction du symbole '\*' qui ne sera pas écrit dans le fichier.
2. Ajouter les instructions qui affichent le nombre de mots ainsi que la longueur moyenne des mots contenus dans le fichier MOTS.TXT.

Ajouter les instructions qui chargent les mots du fichier MOTS.TXT dans un tableau de chaînes de caractères, cherchent les mots de plus de 10 caractères et les écrivent dans un deuxième fichier MOTS10.TXT.

### **Exercice 2:**

Deux fichiers FA et FB dont les noms sont à entrer au clavier contiennent des nombres réels triés dans l'ordre croissant. Écrire un algorithme qui copie le contenu de FA et FB respectivement dans les tableaux TABA et TABB. Ces derniers sont fusionnés dans un troisième tableau trié en ordre croissant TABC. Après la fusion, le tableau TABC est sauvé dans un fichier FC dont le nom est à entrer au clavier.