

*Proposée Par : Mahdhi Mabrouk***Exercice N°1:**

Soit à remplir un Fichier texte **Fe** par des lignes, la saisie se termine en répondant par "N3 à la question "**continuer O/N?**".
Une fois on a rempli le fichier le programme doit copier le contenu du fichier **fe** vers un deuxième fichier **fs** de telle sorte qu'il convertit tout les lettres se trouvant après un point "."

Exp:

On a bac Sc.info, bac Sc.exp.
Fichier texte.fichier des enregistrements.
Algo.pascal.

On a bac Sc.Info, bac Sc.Exp.
Fichier texte.Fichier des enregistrements.
Algo.Pascal.

Exercice N°2:

Soit l'algorithmme suivant:

- 0) **Fonction** scalaire (A,B:tab; n:entire): entire
- 1) $P \leftarrow (A[n]*B[n])$
- 2) **Pour c de n-1 à 0 (pas=-1) faire**
 $P \leftarrow P + (A[c]*B[c])$
 Fin Pour.
- 3) **Scalaire** $\leftarrow P$
- 4) **Fin** Scalaire.

1- Exécuter manuellement cet algorithme avec les paramètres effectifs suivants et indiquer le résultat retourné en donnant chaque fois les valeurs de **c** et **p**. **N=4**

A

| | | | | |
|---|---|---|---|---|
| 3 | 0 | 2 | 7 | 1 |
|---|---|---|---|---|

B

| | | | | |
|---|---|---|---|---|
| 5 | 6 | 7 | 0 | 1 |
|---|---|---|---|---|

2- Donner une analyse, un algorithme en mode récursif réalisant le même traitement.

Exercice N°3:

On dispose d'un fichier texte, intitulé **source.txt**.
Un tautogramme est texte dont les mots commencent par la même lettre (*sans distinction entre majuscule ou minuscule*). **Exemple:** le lion lape le lait lentement. On se propose de sauvegarder toutes les lignes tautogrammes de ce fichier, dans un dans un 2^{ème} fichier intitulé **tauto.txt**

Exercice N°4:

Deux nombres sont des "**Nombres de Mabrouk**", si la somme des chiffres communs de deux nombres est **impaire**. Il s'agit des cas des chiffres communs :

- * Les chiffres communs ont les mêmes positions (C.C.M.P).
- * Les chiffres communs ont des différentes positions (C.C.D.P).

N.S: Somme = (C.C.D.P) + 2*(C.C.M.P);

Exemple: 7195 et 5183 sont des **Nombres de Mabrouk** car la somme = 5 + 2*1 = 7 est impaire.

On se propose d'écrire un programme permettant de remplir aléatoirement un fichier intitulé **nombres.dat** par **N** entiers qui se composent par 4 chiffres où chaque chiffre doit être existé pour une seule fois, de chercher à partir de ce fichier tous deux **Nombres de**



Mabrouk, et les enregistre dans un fichier texte **Nom_Mab.txt** sous la forme suivante:

* Les nombres XXXX et XXXX sont des nombres de Mabrouk.

Exercice N°5:

Ecrire une fonction récursive de **recherche dichotomique** d'un élément dans un tableau d'entiers de taille **n** ordonné par ordre croissant.

Exercice N°6:

Ecrire une fonction récursive qui permet de vérifier si un tableau est symétrie ou non.

Exercice N°7:

Ecrire une fonction récursive qui permet d'inverser un tableau d'entiers.

Exercice N°8:

On se propose de déterminer les coefficients de développement de l'équation $(a+b)^n$:

1. Quelle est la méthode à utiliser pour résoudre ce problème,
2. Ecrire une analyse puis déduire l'algorithme d'un module permettant de déterminer ces coefficients.

Exercice N°9:

On se propose d'écrire un programme qui permet de :

- Remplir une matrice **M** de degré 4 par des entiers **binaires** (0 ou 1 seulement),
- Chaque ligne de la matrice **M** représente la conversion binaire d'un entier **X** de la base 10 ;
 - Trouver la valeur de **X** pour chaque ligne de **M**,
 - Associer les valeurs de **X** dans un tableau **T**,
- Trier puis afficher (en ordre décroissant) les éléments du tableau **T**,
- Enregistrer dans un fichier texte les résultats sous la forme **(X)2 = (Y)10**.

Exemple :

Si M=

| | 1 | 2 | 3 | 4 | | T |
|---|---|---|---|---|---|----|
| 1 | 1 | 0 | 1 | 0 | → | 10 |
| 2 | 0 | 1 | 1 | 1 | → | 7 |
| 3 | 1 | 0 | 0 | 1 | → | 9 |
| 4 | 1 | 0 | 1 | 1 | → | 11 |

$(1010)_2 = (10)_{10}$ comment ?

$$2^3 \quad 2^2 \quad 2^1 \quad 2^0$$
$$1 \quad 0 \quad 1 \quad 0$$

$$1010 = 1*2^3 + 0*2^2 + 1*2^1 + 0*2^0 = 1*8 + 0*4 + 1*2 + 0*1 = 10$$

Le programme affichera : **11-10-9-7**

Questions :

1. Analyser le problème en le décomposant en modules,
2. Analyser chacun des modules envisagés dans l'analyse du programme principal,
3. Déduire de ce qui précède l'algorithme du programme principal ainsi que les algorithmes des modules envisagés.

La vie n'est pas complexe
mais elle a une partie Réelle
et une partie Imaginaire.

