

Objectif : Utilisation des fonctions et procédures.

Exercice 1 : Produit de 2 nombres

Écrire une fonction qui calcule le produit de 2 nombres A et B.

Exercice 2 : Moyenne de 3 nombres

Ecrire un programme se servant d'une fonction MOYENNE pour afficher la moyenne arithmétique de trois nombres réels entrés au clavier.

Exercice 3 : Comparaison de 2 nombres

Ecrire la fonction qui compare deux nombres A et B entre eux et qui détermine si A est inférieur, égal ou supérieur à B.

Exercice 4 : Minimum et Maximum de 4 nombres

Ecrire une fonction MIN et une fonction MAX qui déterminent le minimum et le maximum de deux nombres réels.

Ecrire un programme se servant des fonctions MIN et MAX pour déterminer le minimum et le maximum de quatre nombres réels entrés au clavier.

Exercice 5 : La puissance nième d'un nombre

Ecrire une fonction qui calcule la valeur X^N pour une valeur réelle X (type **double**) et une valeur entière positive N (type **int**) :

Ecrire un programme qui teste la fonction à l'aide de valeurs lues au clavier.

Exercice 6 : Nombre de chiffres

Ecrire la fonction NCHIFFRES du type **int** qui obtient une valeur entière N (positive ou négative) du type **long** comme paramètre et qui fournit le nombre de chiffres de N comme résultat.

Ecrire un petit programme qui teste la fonction NCHIFFRES:

Exemple:

Introduire un nombre entier : 6457392
Le nombre 6457392 a 7 chiffres.

Exercice 7 : Le factoriel

1) Ecrire la fonction qui calcul le factoriel d'un nombre entier positif.

2) Intégrer la fonction définie précédemment dans un programme calculant la somme des factoriels des 100 premiers nombres entiers positif.

Afficher le résultat de la somme des factoriels obtenue.

Les tableaux à une dimension

Exercice 8 : Lecture d'un tableau (1)

Ecrire une fonction LIRE_TAB, à deux paramètres TAB et N, qui permet de remplir un tableau d'entiers.

Exercice 9 : Lecture d'un tableau (2)

La fonction LIRE_TAB à trois paramètres TAB, N et NMAX lit la dimension N et les composantes d'un tableau TAB du type **int**. La dimension N doit être inférieure à NMAX. Implémenter la fonction LIRE_TAB en choisissant bien le type des paramètres.

Exemple:

Pour un appel par : LIRE_TAB(T, N, 10);

La fonction se comportera comme suit:

Dimension du tableau (max.10): 11
Dimension du tableau (max.10): 4
Elément[0] : 43
Elément[1] : 55
Elément[2] : 67
Elément[3] : 79

Exercice 10 : Affichage d'un tableau

Ecrire la fonction ECRIRE_TAB à deux paramètres TAB et N qui affiche N composantes du tableau TAB du type **int**.

Exercice 11 : Somme des éléments d'un tableau

Ecrire la fonction SOMME_TAB qui calcule la somme des N éléments d'un tableau TAB du type **int**. N et TAB sont fournis comme paramètres; la somme est retournée comme résultat du type **long**.

A l'aide des fonctions des exercices précédents, écrire un programme qui lit un tableau A d'une dimension inférieure ou égale à 100 et affiche le tableau et la somme des éléments du tableau.

Exercice 12 : Maximum des éléments d'un tableau

Déterminer le maximum de N éléments d'un tableau TAB d'entiers de deux façons différentes:

- a) la fonction MAX1 retourne la valeur maximale
- b) la fonction MAX2 retourne l'indice de l'élément maximal

Ecrire un programme pour tester les trois fonctions.

Exercice 13 : Fusionner 2 tableaux

Ecrire une procédure qui fusionne deux tableaux d'entiers.

Exercice 14 : Fusionner 3 tableaux

Ecrire une procédure qui fusionne trois tableaux d'entiers.

Exercice 15 : Nombre d'occurrence d'un entier

Ecrire une fonction qui retourne le nombre d'occurrences d'un entier dans un tableau.

Exercice 16 : Première position d'un entier

Ecrire une fonction qui retourne la première position d'un entier dans un tableau.

Exercice 17 : Positions de toutes les occurrences d'un élément

Ecrire une fonction qui retourne toutes positions d'un entier dans un tableau.

Exercice 18 : Recherche, insertion et suppression d'un élément

- 1- Ecrire une procédure qui recherche un élément X dans un tableau d'entiers.
- 2- Ecrire une procédure qui insère un élément X dans une position P d'un tableau d'entiers.
- 3- Ecrire une procédure qui supprime un élément X d'un tableau d'entiers.
- 4- Ecrire la fonction INSERER qui place un élément X à l'intérieur d'un tableau qui contient N éléments triés par ordre croissant, de façon à obtenir un tableau à N+1 éléments triés par ordre croissant. La dimension du tableau est incrémentée dans la fonction INSERER.
- 5- Ecrire un programme qui teste les fonctions définies plus haut.

Exercice 19 : Tri des éléments

Ecrire un programme utilisant 4 fonctions :

- Une fonction de lecture d'un tableau,
- Une fonction de tri par sélection de ce tableau,
- Une fonction de tri à bulles de ce tableau,
- Une fonction d'affichage du tableau trié.

Les Chaines de caractères**Exercice 20 : Longueur d'une chaîne**

Ecrire la fonction LONG_CH qui retourne la longueur d'une chaîne de caractères CH comme résultat.

Exercice 21 : Chaîne palindrome

Ecrire une fonction qui teste si une chaîne de caractères est palindrome.

Exercice 22 : Comparaison de 2 chaînes

Ecrire une fonction qui compare deux chaînes de caractères (même principe que strcmp).

Exercice 23 : Conversion en Majuscule

Ecrire la fonction MAJ_CH qui convertit toutes les lettres d'une chaîne en majuscules, sans utiliser de variable d'aide.

Exercice 24 : Copie d'une chaîne à la fin d'une autre

Ecrire la fonction AJOUTE_CH à deux paramètres CH1 et CH2 qui copie la chaîne de caractères CH2 à la fin de la chaîne CH1 sans utiliser de variable d'aide.

Exercice 25 : Inverse d'une chaîne

Ecrire la fonction INVERSER_CH qui inverse l'ordre des caractères d'une chaîne CH. Utiliser la fonction LONG_CH (définie plus haut) et définir une fonction d'aide PERMUTER_CH qui échange les valeurs de deux variables du type **char**.

Exercice 26 : Nombre de mots

Ecrire la fonction NMOTS_CH qui retourne comme résultat le nombre de mots contenus dans une chaîne de caractères CH.

Les tableaux à deux dimensions**Exercice 27 : Lecture d'une matrice**

Ecrire la fonction LIRE_MATRICE qui lit les éléments d'une matrice MAT du type **int** et de dimensions L et C.

Exercice 28 : Affichage d'une matrice

Ecrire la fonction AFFICHER_MATRICE qui affiche les éléments de la matrice de dimensions L et C.

Exercice 29 : Somme des éléments d'une matrice

Ecrire la fonction SOMME_MATRICE du type **long** qui calcule la somme des éléments d'une matrice MAT du type **int**. Choisir les paramètres nécessaires. Ecrire un petit programme qui teste la fonction SOMME_MATRICE.

Exercice 30 : Addition de deux matrices

Ecrire la fonction ADDITION_MATRICE qui effectue l'addition des matrices suivante:

$$\text{MAT1} = \text{MAT1} + \text{MAT2}$$

Choisir les paramètres nécessaires et écrire un petit programme qui teste la fonction ADDITION_MATRICE.

Exercice 31 : Produit de deux matrices

Ecrivez un programme qui réalise le produit des 2 matrices :

Utilisez 2 fonctions : PRODUIT_MATRICE pour calculer le produit et ECRIRE_MATRICE pour l'afficher.

Exercice 32 : Minimum de chaque ligne et chaque colonne

Ecrire un programme en langage C, contenant 4 fonctions réalisant les questions F1, F2, F3 et F4, qui permet de :

F1. Lire une matrice de N lignes et M colonnes ligne par ligne. Les données sont dans l'ordre :

F2. Ecrire ensuite cette matrice ligne par ligne.

F3. Renvoyer le minimum de la matrice

F3. Afficher le minimum de chaque ligne.

F3. Afficher le minimum de chaque colonne.

F4. Afficher le numéro de la colonne contenant le minimum de la matrice. (A égalité, donner le plus petit numéro de colonne).

Ecrire un programme qui teste chacune des fonctions demandées ci-dessous (le programme main()).