

Développer l'organigramme et le programme permettant :

- Compter dans un tableau d'entiers les nombres étant supérieurs à une valeur donnée.
- Compter dans un tableau d'entiers les nombres compris dans une fourchette donnée (supérieur à X et inférieur à Y).
- Fournir la somme des nombres présents dans un tableau d'entiers.
- Soit un tableau d'entiers dont vous devez demander à l'utilisateur l'introduction au clavier de chacune des valeurs. (Une demande séparée par valeur).
- Soit un tableau d'entiers (non trié) dans lequel vous devez rechercher le nombre le plus grand.
- Soit un tableau d'entiers dans lequel vous devez rechercher la présence d'un nombre donné. (Exemple : rechercher si le nombre 12 se trouve dans le tableau).
- Soit une chaîne de caractères que vous devez convertir en majuscule. Attention car certains caractères peuvent déjà être en majuscule tandis que d'autres caractères ne sont pas convertibles. Il faut vous baser sur les codes ascii (A :65 -> Z :90) (a :97->z :122)

Suite des exercices.

- Soit un tableau à deux dimensions représentant une matrice X lignes et Y colonnes. Prévoir l'organigramme et le code permettant l'introduction par l'utilisateur de chacun des éléments de la matrice ligne par ligne et pour chaque ligne, colonne par colonne.

Exemple :

introduire mat[0][0]

introduire mat[0][1]

introduire mat[0][2]

introduire mat[1][0] ...etc

introduire mat[3][2]

La matrice est de dimension [4][3]

- Soit un tableau d'entiers. Vérifier que le tableau est bien trié par ordre croissant c.-à-d. que nombre suivant dans le tableau est bien plus grand que le nombre précédent (correspondant à l'indice juste inférieur)

- Soit un tirage lotto. Nous retrouvons un tirage de 7 nombres de façon aléatoire, chaque nombre étant unique et compris entre 1 et 42. Pour chacun des nombres tirés aléatoirement, il faudra simplement vérifier que le nombre n'a pas été tiré au préalable.

- Vérifier qu'un nombre introduit au clavier par l'utilisateur est un nombre premier ou pas. Tout nombre premier n'est divisible entièrement (sans laisser de reste) que par 1 et par lui-même.

- Prévoir le code permettant d'inverser une chaîne de caractères. Exemple : si la chaîne de départ vaut MAISON, le résultat de l'exécution de votre programme sur cette chaîne donnera donc NOSIAM. Deux versions du même code peuvent être envisagées : l'une acceptant un

deuxième tableau (version la plus simple) et l'autre modifiant directement le tableau de départ (avec donc un seul tableau utilisé).

- Ecrire un programme qui calcule le produit scalaire de deux vecteurs d'entiers U et V (de même dimension).

Soit un vecteur I 10 14 2 5

Soit un vecteur J 6 9 4 7

Le produit scalaire de ces deux vecteurs va donc donner : $(10*6 + 14*9 + 2*4 + 5*7)$

-Un tableau A de dimension N+1 contient N valeurs entières triées par ordre croissant; la (N+1)^{ième} valeur est indéfinie. Insérer une valeur VAL donnée au clavier dans le tableau A de manière à obtenir un tableau de N+1 valeurs triées.

Vous pouvez essayer de refaire l'exercice sur la recherche d'un élément dans un tableau d'entiers trié. Cette recherche se fera par méthode dichotomique. Si ce n'est pas fait, ce n'est pas grave.

Suite des exercices (déclarations et définitions des fonctions).

Pour chacun des exercices précédents, nous allons envisager leur intégration dans le corps d'une fonction. Il n'est pas nécessaire de définir dans cette première démarche toute la fonction mais se limiter à la déclaration du prototype de la fonction.

a) - *Compter dans un tableau d'entiers les nombres compris dans une fourchette donnée (supérieur à X et inférieur à Y).*

Les entiers X et Y ainsi que le tableau d'entiers doivent être passés comme arguments à la fonction tandis que le résultat du comptage sera renvoyé à la fonction appelante comme valeur de retour.

b) *Soit un tableau d'entiers (non trié) dans lequel vous devez rechercher le nombre le plus grand.*

Le tableau d'entier est passé comme argument tandis que le nombre trouvé doit être renvoyé vers la fonction appelante comme valeur de retour.

c) *Soit un tableau d'entiers dans lequel vous devez rechercher la présence d'un nombre donné. (Exemple : rechercher si le nombre 12 se trouve dans le tableau).*

Le tableau d'entier et l'entier recherché sont passés comme argument tandis que l'indication de la recherche fructueuse ou pas se fait par la valeur de retour sous forme d'un booléen (BOOL).

Valeur retournée TRUE pour indiquer que le nombre recherché se trouve dans le tableau
Valeur retournée FALSE pour indiquer que le nombre recherché ne se trouve pas dans le tableau.

d) - *Vérifier qu'un nombre introduit au clavier par l'utilisateur est un nombre premier ou pas. Tout nombre premier n'est divisible entièrement (sans laisser de reste) que par 1 et par lui-même.*

Ecrire une fonction qui aura comme seul objectif de vérifier que l'entier passé en argument est un nombre premier ou pas. Cette information sera renvoyée par la fonction par valeur de retour au moyen d'un booléen : TRUE -> le nombre est premier, FALSE -> le nombre n'est pas premier.

e) Pour chacune des fonctions reprises ci-dessus, donner un petit exemple par lequel vous mettrez en évidence l'utilisation de cette fonction lors de son appel avec les passages de paramètres.

f) Dans le prototype de la fonction malloc, nous retrouvons un type de retour sous la forme void * malloc ...

Veillez mettre en évidence l'utilisation d'un tel type de retour.

g) Ecrire complètement (prototype et définition) d'une fonction permettant de calculer la moyenne sur un tableau d'entiers. Le résultat de cette moyenne calculée est renvoyé la fonction appelante par valeur de retour. A vous à déterminer les arguments dont vous aurez besoin.

h) Ecrire une fonction récursive permettant de calculer le factoriel d'un nombre.

$$n! = n * (n-1) * (n-2) * \dots * 3 * 2 * 1$$

i) Ecrire une fonction permettant de permuter le contenu de deux variables de type char.

Suite des exercices.

j) Soit une fonction devant permettre le calcul d'un polynôme de degré N

$$P(X) = A_n X^n + A_{n-1} X^{n-1} + \dots + A_1 X + A_0$$

Un tableau de flottant contenant les différents coefficients ($A_n \dots A_0$) ainsi que la valeur X seront passés en argument à la fonction tandis que le résultat du calcul sera renvoyé à la fonction appelante par valeur de retour.

Utilisez le *schéma de Horner* qui évite les opérations d'exponentiation:

$$\begin{array}{l} \underbrace{A_n}_{\text{}} \\ \underbrace{* X + A_{n-1}}_{\text{}} \\ \underbrace{* X + A_{n-2}}_{\text{}} \\ \dots \\ \underbrace{* X + A_0}_{\text{}} \end{array}$$

k) Soit un tableau d'entiers comprenant des nombres positifs, négatifs ou nuls. Ecrire un programme permettant de supprimer de cette liste les nombres nuls en faisant remonter les nombres suivants (d'indices supérieurs) dans les cases ainsi libérées. Il faut travailler sur le même tableau

Exemple : avant traitement... 1 2 3 0 4 5 8 0 9

Après traitement : 1 2 3 4 5 8 9

l) Ecrivez un programme qui, étant donnés deux tableaux de nombres entiers A et B, de même longueur, calcule et imprime le nombre de valeurs de i pour lesquelles on a $A_i = B_i$

m) Ecrire un programme qui affiche le pgcd de deux entiers entrés au clavier. Vous devez pour cela utiliser l'algorithme d'Euclide.

L'algorithme d'Euclide permet de calculer le P.G.C.D. de deux entiers naturels a et b tels que $a > b$.

Il consiste à réitérer les manipulations suivantes :

- Effectuer la division euclidienne de a par b. Soit r le reste.
- Remplacer a par b et b par r. On a $b > r$ d'après la définition de la division euclidienne.

Le P.G.C.D. est le dernier reste non nul. (Si b divise a, le P.G.C.D. est b)

n) En travaillant avec la technique d'allocation dynamique :

1. Créer un tableau d'entiers T de taille n où n est entré au clavier. Initialiser chacun de ses éléments à une valeur aléatoire.
2. Créer un tableau de taille $n+1$ correspondant au tableau précédent auquel on a ajouté un élément saisi au clavier.
3. Remplacer le premier tableau T par le tableau à $n+1$ éléments créé à l'étape précédente. (En travaillant sur le contenu des pointeurs et en n'oubliant pas de libérer l'espace utilisé par le premier tableau).

o) Ecrire la fonction MULTI_MATRICE qui effectue la multiplication d'une matrice (tableau à deux dimensions, k lignes, p colonnes) par un entier X:

$$M_{i,j} = X * M_{i,j} \text{ (} i \text{ allant de 0 jusque } k-1 \text{ et } j \text{ allant pour chaque } i \text{ de 0 jusque } p-1 \text{)}$$

Ecrire le programme permettant de tester votre fonction.