

TP Initiation à la programmation

Série de TP N°1 – Tableaux à une dimension - Vecteurs

But de TP :

Ce TP a pour objectif d'initier les étudiants à la déclaration, à l'initialisation et à la manipulation des tableaux à une dimension (vecteurs), à travers des exercices pratiques permettant de consolider les notions fondamentales de programmation.

Exercice N°01 : Soit l'algorithme suivant :

Algorithme Vecteur ;

Variables

V : Tableau [0..99] d'entier ;

N, i : entier ;

début

// Entrées

Écrire("Donner la taille du vecteur V : ") ;

Lire(N) ;

Écrire("Donner les éléments du vecteur V : ") ;

Pour i ← 0 à N-1 faire

 Lire(V[i]) ;

Fin-Pour ;

// Traitements

Pour i ← 0 à N-1 faire

 V[i] ← V[i] * 5 ;

Fin-Pour ;

//sorties

Écrire("le vecteur V = ") ;

Pour i ← 0 à N-1 faire

 Écrire(V[i]) ;

Fin-Pour ;

Fin.

Questions :

1. Traduire l'algorithme en un programme en langage C.
2. Compiler et exécuter le programme pour les valeurs suivantes : N = 4 et V = [6 8 5 12] ;
3. Dérouter le Prog-C pour les valeurs de N et T ci-dessus.
4. Dédire ce que fait cet algorithme.
5. Réécrire le programme en remplaçant la boucle **for** par la boucle **While** uniquement dans la partie **traitements**.

Exercice N°02 :

Écrire un programme en langage C qui permet de saisir un vecteur **V** = [1 3 5 7 9] de nombres réels, puis de calculer et d'afficher la **somme** et la **moyenne** de ses éléments. Ensuite, améliorer le programme afin de calculer également la **variance** σ^2 et l'**écart-type** σ du vecteur, et d'afficher tous ces résultats. **Rappel :** $\sigma^2 = \frac{1}{n} \sum_{i=0}^{n-1} (V_i - moy)^2$.

Exercice N°03 :

On considère **n** positions **y[i]** mesurées à des intervalles de temps constants **dt**. Écrire un programme en langage C qui permet de : saisir les **n** valeurs du vecteur des positions **y** ensuite qui calcule le vecteur vitesse **V** entre deux mesures successives, selon la relation : $v_i = (y_{i+1} - y_i) / dt$.

Compiler et exécuter le programme pour n=6 , dt=2 et V= [1 4 9 19 38 72]

Exercice N°04 :

Écrire un programme en langage C permettant de calculer le produit scalaire de deux vecteurs ayant la même dimension.

Compiler et exécuter le programme pour n=5 A= [1 3 2 5 10], B= [1 2 3 4 5]

Exercice N°05 :

Écrire un programme en langage C permettant de saisir **n** éléments d'un vecteur T, puis de déterminer et d'afficher le nombre d'éléments **positifs (nPos)**, **négatifs (nNeg)** et **nuls (nNul)** qu'il contient.

Compiler et exécuter le programme pour n=6 et T= [1 0 -5 0 9 20]

TP Initiation à la programmation

Exercices supplémentaires ✎ :

Exercice Sup-01 :

Écrire un programme en langage C qui permet de saisir un vecteur V contenant N nombres réels, ainsi qu'une valeur réelle X. Le programme devra ensuite vérifier si la valeur X appartient au vecteur. Si X est trouvée, son indice sera affiché ; dans le cas contraire, le message suivant sera affiché : « **La valeur X n'existe pas dans V.** »

Exercice Sup-02:

Écrire un programme en langage C permettant de saisir un vecteur V de N nombres réels, où N est un entier fourni par l'utilisateur. Le programme devra ensuite déterminer et afficher la valeur **maximale** et la valeur **minimale** parmi les éléments du vecteur.

Exercice Sup-03:

Soit V un vecteur de type réel de taille N. Écrire un algorithme/programme C qui permet de trier (d'ordonner) les éléments du vecteur V avec un ordre croissant puis décroissant.

Exercice Sup-04:

Écrire un programme en langage C qui permet de saisir deux vecteurs A et B de tailles respectives N et M. Le programme devra fusionner ces deux vecteurs dans un troisième vecteur C en conservant l'ordre des éléments : d'abord tous les éléments de A dans l'ordre, puis tous les éléments de B dans l'ordre. Enfin, le programme affichera le vecteur fusionné et sa taille totale. Supposons : Vecteur A : [1 3 5] et Vecteur B : [2 4 6 8]

Fusion en C (ordre A puis B) : Vecteur C résultant : [1 3 5 2 4 6 8]

Exercice Sup-05:

Écrire un programme en langage C qui permet de saisir un vecteur contenant 5 éléments. Le programme devra ensuite supprimer l'élément situé à la **position 3** du vecteur, ajuster les éléments restants en les décalant pour conserver leur ordre, et afficher le vecteur résultant ainsi que sa **nouvelle taille**. Si la position spécifiée est invalide, le programme devra afficher un message d'erreur.

Exercice Sup-06:

Écrire un programme en langage C qui permet de saisir un vecteur de N entiers, puis d'inverser l'ordre de ses éléments, de manière à ce que le premier élément devienne le dernier, le deuxième devienne l'avant-dernier, et ainsi de suite. Le programme devra ensuite afficher le vecteur inversé.

Une extension possible consiste à vérifier si le vecteur est symétrique (palindrome), c'est-à-dire si le vecteur reste identique après inversion, et à compter le nombre d'éléments qui restent à la même position après l'inversion.