
Rattrapage, 19 Janvier 2018, Durée : 1h30

Exercice 1 :

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) `A+3` d) `P+(*P-10)` e) `&A[7]-P` f) `*(P+*(P+8)-A[7])`

Exercice 2 :

Deux nombres entiers n et m sont qualifiés d'amis, si la somme des diviseurs de n est égale à m et la somme des diviseurs de m est égale à n (on ne compte pas comme diviseur le nombre lui-même et 1).

Exemple : 48 et 75 sont deux nombres amis car $2+3+4+6+8+12+16+24=75$ et $3+5+15+25=48$.

1. Écrivez une fonction qui permet de calculer la somme des diviseurs d'un entier n en excluant 1 et n.
2. Écrivez une fonction qui permet de déterminer si deux entiers n et m sont amis ou non.
3. En utilisant la fonction de la question précédente, écrivez un programme qui calcule et affiche le nombre de nombres amis se trouvant entre 1 et 1000.

Exercice 3 :

Écrivez un programme C qui :

1. Saisit les éléments d'un tableau T de taille 10 de type float, ainsi que les éléments d'une matrice carrée A de taille (10*10) de type float.
2. Calcule les éléments du vecteur V qui est égale à la multiplication du tableau T et de la matrice A.
3. Saisit une valeur réelle x et affiche l'indice de la dernière colonne qui contient x s'il se trouve dans la matrice, sinon le message que x n'est pas dans la matrice.

Exercice 4 :

Remarque : Dans cet exercice, utilisez uniquement des pointeurs pour parcourir les tableaux

Écrivez un programme C qui :

1. Saisit les dimensions n et m de deux tableaux d'entiers A et B, alloue la mémoire dynamiquement à ces tableaux, puis saisit leurs éléments.
2. Ajoute les éléments de B qui ne se trouvent pas dans A à la fin du tableau A.

Exemple :

Pour le tableau A :

9	2	6	4	8
---	---	---	---	---

et le tableau B :

4	3	2	10
---	---	---	----

Le nouveau tableau A devient :

9	2	6	4	8	3	10
---	---	---	---	---	---	----