


I'm not robot  reCAPTCHA

I am not robot!

Algorithme les tableaux exercices corriges pdf

EXERCICES CORRIGÉS EN ALGORITHMIQUE : LES TABLEAUX Cliquer au dessous pour le téléchargement

Exercice 1 : Ecrire un algorithme permettant d'entrer cinq valeurs réelles au clavier, les stocker dans un tableau, calculer leur somme et les afficher avec leur somme à l'écran. Correction exercice 1 : Algorithme tableau somme. Var V : tableau [1..5] de réels ; S : réel ; i : entier ; Debut (*lecture des éléments du tableau*) Lire(T[i]) ; Finpour i (calcul de la somme des éléments du tableau) S <- 0 ; Pour i <- 1 à 5 faire S <- S + V[i] ; Finpour i (*affichage des éléments du tableau*) Pour i <- 1 à 5 faire Ecrire('l'élément N° ',i,' est : ',V[i]) ; Finpour i Ecrire('la somme des éléments du tableau est : ',S) ; fin

Exercice 2 : Ecrire un algorithme permettant de saisir et d'afficher N éléments d'un tableau. Correction exercice 2 : Algorithme saisie affichage. Var T : tableau [1..100] de réels ; N,i : réel ; Debut Ecrire('entrer le nombre d'éléments du tableau : '); Lire(N) ; (*lecture des éléments du tableau*) Pour i <- 1 à N faire Ecrire('entrer l'élément N° ',i) ; Lire(T[i]) ; Finpour i (*affichage des éléments du tableau*) Pour i <- 1 à N faire Ecrire('l'élément T[',i,'] est : ',T[i]) ; Finpour i Fin Exercice 3 : Ecrire un algorithme permettant de calculer la somme, produit et moyenne des éléments d'un tableau. Correction exercice 3 : Algorithme somme produit moyenne. Var T : tableau [1..100] de réels ; N,i : entiers ; S,P,M : réels ; Debut Si N=0 alors Ecrire('le tableau est vide') ; Sinon S <- 0 ; P <- 1 ; Moyenne <- 0 ; Pour i <- 1 à N faire Moyenne <- Moyenne + T[i] ; Finpour i Moyenne <- Moyenne / N ; Ecrire('la somme des éléments du tableau est : ',S) ; Ecrire('le produit des éléments du tableau est : ',P) ; Ecrire('la moyenne des éléments du tableau est : ',M) ; Finsi Fin Exercice 4 : Ecrire un algorithme permettant de consulter un élément d'un tableau. Correction exercice 4 : Algorithme consultation. Var T : tableau [1..100] de réels ; N,P : entiers ; Debut Si N=0 alors Ecrire('le tableau est vide') ; Sinon Lire(P) ; Ecrire('entrer l'indice de l'élément à consulter : '); Lire(N) ; Ecrire('la valeur de l'élément à consulter est : ',T[N]) ; Fin

Exercice 5 : Ecrire un algorithme permettant de chercher toutes les occurrences d'un élément dans un tableau. Correction exercice 5 : Algorithme recherche toutes occurrences. Var T : tableau [1..100] de réels ; N,i : entiers ; X : réel ; Existe : booléen ; Debut Si N=0 alors Ecrire('le tableau est vide') ; Sinon Existe <- Vrai ; Ecrire('l'élément à chercher apparait à la position : ',i) ; Finsi Finpour i Si Existe = Faux alors Ecrire('la première occurrence de l'élément dans ce tableau est : ',P) ; Sinon Existe <- Vrai ; Ecrire('la première occurrence de l'élément dans ce tableau est : ',P) ; Finsi

Exercice 6 : Algorithme recherche première occurrence ; Var T : tableau [1..100] de réels ; P,N,i : entiers ; X : réel ; Existe : booléen ; Debut Si N=0 alors Ecrire('le tableau est vide') ; Sinon Existe <- Vrai ; Ecrire('la première occurrence de l'élément dans ce tableau est : ',P) ; Finsi

Exercice 7 : Algorithme recherche dernière occurrence ; Var T : tableau [1..100] de réels ; P,N,i : entiers ; X : réel ; Existe : booléen ; Debut Si N=0 alors Ecrire('le tableau est vide') ; Sinon Existe <- Vrai ; Ecrire('la dernière occurrence de l'élément dans ce tableau est : ',P) ; Finsi

Exercice 8 : Algorithme fréquence de l'élément dans ce tableau est : ',P) ; Var T : tableau [1..100] de réels ; N,i,F : entiers ; X : réel ; Debut Si N=0 alors Ecrire('le tableau est vide') ; Sinon F <- 0 ; Pour i <- 1 à N faire F <- F + T[i] ; Finpour i F <- F / N ; Ecrire('la fréquence de l'élément dans ce tableau est : ',F) ; Fin

Exercice 9 : Algorithme Ajout. Var T : tableau [1..100] de réels ; N : entiers ; X : réel ; Rep : caractère ; Debut Ecrire('entrer la valeur de l'élément à ajouter : '); Lire(X) ; Ecrire('Confirmer l'ajout (O/N) : '); Lire(Rep) ; Si Rep = 'O' alors (*la valeur 'o' pour 'oui' et 'N' pour 'Non') Ecrire('l'élément apparait : ', F, 'fois dans ce tableau') ; Finsi

Exercice 10 : Algorithme Modification. Var T : tableau [1..100] de réels ; N,P : entiers ; X : réel ; Rep : caractère ; Debut Si N=0 alors Ecrire('le tableau est vide') ; Sinon Ecrire('entrer l'indice de l'élément à modifier : '); Lire(P) ; Ecrire('Position hors limites du tableau') ; Sinon Ecrire('Position hors limites du tableau') ; Finsi

Exercice 11 : Ecrire un algorithme permettant d'insérer un élément dans un tableau (au début, au milieu ou à la fin). Correction exercice 11 : Algorithme insertion. Var T : tableau [1..100] de réels ; N,P,i : entiers ; X : réel ; Rep : caractère ; Debut Si N=0 alors Ecrire('le tableau est vide') ; Sinon Ecrire('entrer la valeur de l'élément à insérer : '); Lire(X) ; Ecrire('Entrer la position d'insertion : '); Lire(P) ; Ecrire('Position hors limites du tableau') ; Sinon Ecrire('Confirmer l'insertion (O/N) : '); Lire(Rep) ; Si Rep = 'O' alors N <- N + 1 ; Pour i <- N à P+1 Faire T[i] <- T[i-1] ; Finpour i T[P] <- X ; Ecrire('le tableau est vide') ; Finsi

Exercice 12 : Ecrire un algorithme permettant de supprimer un élément dans un tableau. Correction exercice 12 : Algorithme suppression. Var T : tableau [1..100] de réels ; N,P,i : entiers ; X : réel ; Rep : caractère ; Debut Si N=0 alors Ecrire('le tableau est vide') ; Sinon Rep : caractère ; Debut Si N=0 alors Ecrire('la valeur dans cette position est : ',T[P]) ; Finsi

Exercice 13 : Ecrire un algorithme permettant de trier par ordre croissant les éléments d'un tableau. Correction exercice 13 : Algorithme tri_Croissant. Var T : tableau [1..100] de réels ; N,i,j : entiers ; Aux : réel ; Debut Si N=0 alors Ecrire('le tableau est vide') ; Sinon Pour i <- 1 à N-1 Faire Si T[i] > T[i+1] alors Aux <- T[i] ; T[i] <- T[i+1] ; T[i+1] <- Aux ; Finpour i

Exercice 14 : Ecrire un algorithme permettant de trier par ordre décroissant les éléments d'un tableau. Correction exercice 14 : Algorithme tri_Décroissant. Var T : tableau [1..100] de réels ; N,i,j : entiers ; Aux : réel ; Debut Si N=0 alors Ecrire('le tableau est vide') ; Sinon Pour i <- 1 à N-1 Faire Pour j <- i+1 à N Faire Si T[i] < T[j] alors T[i] <- T[j] ; T[j] <- Aux ; Aux <- T[i] ; Finpour j Finpour i

Exercice 15 : Ecrire un algorithme permettant de fusionner les éléments de deux tableaux T1 et T2 dans un autre tableau T. N.B : N : nombre des éléments du tableau T1 M : nombre des éléments du tableau T2 Correction exercice 15 : Algorithme fusion deux tableaux. Var T1,T2 : tableau [1..100] de réels ; T : tableau [1..200] de réels ; N,M,i : entiers ; Debut Si (N=0) et (M=0) alors Ecrire('le tableau est vide') ; Sinon Pour i <- 1 à M Faire T[i] <- T1[i] ; Finpour i Pour i <- 1 à N Faire T[i+M] <- T2[i] ; Finpour i

Exercice 16 : Algorithme tableau deux dimension. Var T : tableau [1..10,1..4] de réels ; I,j : entiers ; S,P,M : réels ; Debut (*saisie des éléments du tableau*) Pour i <- 1 à 10 Faire Pour j <- 1 à 4 Faire S <- S + T[i,j] ; P <- P + T[i,j] ; Finpour j Finpour i M <- S/40 ; (*40 : nombre d'élément du tableau = 10x4 *) Ecrire('la somme des éléments du tableau est : ',S) ; Ecrire('la moyenne des éléments du tableau est : ',M) ; Fin

Exercice 17 : Algorithme diagonale de matrice. Const N=8 ; Var M : tableau [1..8,1..8] d'entiers ; I,j : entiers ; Sdiag : entiers ; Debut (*saisie des éléments de la matrice*) Sdiag <- 0 ; Pour i <- 1 à N Faire Ecrire('l'élément M[',i,',' ,j,'] : '); Lire(M[i,j]) ; Sdiag <- Sdiag + M[i,j] ; Finpour i Ecrire('la somme des éléments de la diagonale est : ',Sdiag) ; Fin

Exercice 18 : Ecrire un algorithme permettant d'effectuer le produit des matrices A(n,m) et B(m,p). n, m et p données (par exemple n=4,m=5,p=3). N.B : Pour pouvoir faire le produit de deux matrices, il faut absolument que le nombre de colonnes de la première soit égal au nombre de lignes de la deuxième. Correction exercice 18 : Algorithme produit matrices. Const n=4 ; m=5 ; p=3 ; Var A : tableau [1..n,1..p] de réels ; B : tableau [1..p,1..m] de réels ; C : tableau [1..n,1..m] de réels ; I,j : entiers ; Debut (*lecture des éléments des deux matrices*) Pour i <- 1 à n Faire Pour j <- 1 à p Faire Lire(A[i,j]) ; Lire(B[i,j]) ; Finpour j Finpour i C[i,j] <- 0 ; Pour i <- 1 à p Faire C[i,j] <- C[i,j] + A[i,k] * B[k,j] ; Finpour k Finpour i

Exercice 19 : Ecrire un algorithme permettant de construire dans une matrice carrée P et d'afficher le triangle de PASCAL de degré N. N.B : On pourra utiliser cette relation pour les éléments de triangle de PASCAL. P[i,j] = P[i-1,j-1] + P[i-1,j] Exemple : triangle de pascal de degré 5 : N=0 1 N=1 1 1 N=2 1 2 1 N=3 1 3 3 1 N=4 1 4 6 4 1 N=5 1 5 10 10 5 1 Correction exercice 19 : Algorithme triangle pascal. Var P : tableau [1..100,1..100] de réels ; I,j,n : entiers ; Debut Ecrire('entrer l'ordre du triangle de pascal que vous voulez : '); Lire(n) ; (*remplissage du triangle de Pascal*) P[1,1] <- 1 ; Pour i <- 2 à n+1 Faire Pour j <- 1 à i Faire Ecrire(P[i,j], ' '); Finpour j (*retour à la ligne *) Finpour i