

Matière : INFORMATIQUE 2 : Algorithmique

ST : Série 3

NE PAS FAIRE ATTENTION A CE CADRE VIDE

NE PAS FAIRE ATTENTION A CE CADRE VIDE

3° : Ecrire un algorithme qui déclare, remplit et affiche un tableau de N valeurs numériques, tel que $N \leq 15$.

Corrigé

- Ce premier exercice sur les vecteurs montre comment remplir un vecteur ou un tableau et comment l'afficher en utilisant une autre boucle.
- Comme l'exercice stipule que $N \leq 15$, donc on déclare un tableau de 15 éléments.
- Ne pas oublier d'insister sur le fait que l'indice du tableau commence toujours par « 0 ». Dans notre exemple l'indice varie de 0 à N-1.

```

Algorithme Vecteur_1 ;
Var
    N , i : Entier ;
    V : Tableau [ 15 ] de Réel ;
Début
    Ecrire ( ' Donnez le nombre de valeurs ' ) ;
    Lire ( N ) ;
    // Lecture et Remplissage du tableau (ou vecteur) //
    Pour i : = 0 à N - 1 faire
        Début
            Ecrire ( ' Donnez la valeur de V [ ' , i , ' ] ' ) ;
            Lire ( V [ i ] ) ;
        Fin ;
    // Affichage du tableau //
    Pour i : = 0 à N - 1 faire
        Ecrire ( ' V [ ' , i , ' ] = ' , V [ i ] ) ;
Fin.

```

4° : Écrire l'algorithme qui remplit un tableau avec la table de multiplication par 9.

Corrigé

```

Algorithme mult_9 ;
Var
    N , i : Entier ;
    V : Tableau [ 11 ] de Entier ;
Début
    // Remplissage du tableau //
    Pour i : = 0 à 10 faire
        V [ i ] : = 9 * i ;
    // Affichage du tableau //
    Pour i : = 0 à 10 faire
        Ecrire ( ' 9 * ' , i , ' = ' , V [ i ] ) ;
Fin.

```

Il est possible de regrouper les deux boucles dans une seule boucle

```

Algorithme mult_9 ;
Var
    N , i : Entier ;
    V : Tableau [ 11 ] de Entier ;
Début
    // Remplissage et affichage du tableau //
    Pour i : = 0 à 10 faire
        Début
            V [ i ] : = 9 * i ;
            Ecrire ( ' 9 * ' , i , ' = ' , V [ i ] ) ;
        Fin ;
Fin.

```

5° : Que produit l'algorithme suivant ?

Corrigé

Cet algorithme enregistre le carré des nombres de 0 à 9 dans un vecteur.

```

Algorithme Vecteur_2 ;
Var
    i : Entier ;
    V : Tableau [ 10 ] de Entier ;
Début
    // Remplissage du tableau //
    Pour i : = 0 à 9 faire
        V [ i ] : = i * i ;
    // Affichage du tableau //
    Pour i : = 0 à 9 faire
        Ecrire ( i , ' * ' , i , ' = ' , V [ i ] ) ;
Fin.

```

Il est possible de regrouper les deux boucles dans une seule boucle

```

Algorithme Vecteur_2 ;
Var
    i : Entier ;
    V : Tableau [ 10 ] de Entier ;
Début
    // Remplissage et affichage du tableau //
    Pour i : = 0 à 9 faire
        Début
            V [ i ] : = i * i ;
            Ecrire ( i , ' * ' , i , ' = ' , V [ i ] ) ;
        Fin ;
Fin.

```

6° : Ecrire un algorithme qui remplit un tableau de la moyenne d'un groupe composé, au maximum, de 30 étudiants, puis calcule la moyenne du groupe.

Corrigé

Ici, on pousse l'étudiant à mieux comprendre comment manipuler l'indice ainsi qu'un élément d'un vecteur.

```

Algorithme Moyenne ;
Var
    N , i : Entier ;
    V : Tableau [ 30 ] de Réel ;
    S, Moy : Réel ;
Début
    Ecrire ( ' Donnez le nombre des étudiants N ' ) ;
    Lire ( N ) ;
    // Remplissage du tableau (ou vecteur) et calcul de la somme des moyennes //
    S : = 0 ;
    Pour i : = 0 à N - 1 faire
        Début
            Ecrire ( ' Donnez la note du ' , i + 1 , ' °étudiant : ' ) ;
            Lire ( V [ i ] ) ;
            S : = S + V [ i ] ;
        Fin ;
    // Calcule et Affichage de la moyenne du groupe //

```

```

Moy : = S / N ;
Ecrire ( ' Moyenne du groupe = ' , Moy ) ;
Fin.

```

7° : Ecrire l'algorithme qui remplit un tableau avec la suite logique :
3, 8, 13, 18, 23, 28,, 88, 93, 98, 100

Corrigé

Cette suite à une progression de 5, sauf pour la dernière valeur « 100 »
On poussera l'étudiant à trouver l'astuce pour afficher la dernière valeur.

```

Algorithme suite_1 ;
Var
    i , X : Entier ;
    V : Tableau [ 21 ] de Entier ;
Début
    // On démarre avec la valeur 3 //
    X : = 3 ;
    // On remplit la suite de 8 à 98 avec un pas de 5 //
    Pour i : = 0 à 19 faire
        Début
            V [ i ] : = X ;
            X : = X + 5 ;
        Fin ;

    // la dernière valeur « 100 » est introduite à la fin de la boucle//
    V [ 20 ] : = 100 ;

    Pour i : = 0 à 20 faire
        Ecrire ( ' V [ ' , i , ' ] = ' , V [ i ] ) ;
Fin.

```

8° : Que produit l'algorithme suivant ?

Corrigé

Cet algorithme va afficher les nombres impairs comme suit :
1, 3, 5, 7, 9, 11, 13, 15, 17, 19.

```

Algorithme Impair ;
Var
    i : Entier ;
    V : Tableau [ 10 ] de Entier ;
Début
    // Remplissage du tableau //
    V [ 0 ] : = 1 ;
    Pour i : = 1 à 9 faire
        V [ i ] : = V [ i - 1 ] + 2 ;

    // Affichage du tableau //
    Pour i : = 0 à 9 faire
        Ecrire ( V [ i ] , ' , ' ) ;

    // A cause du « . » Le dernier élément est affiché à la fin de la boucle //
    Ecrire ( V [ 9 ] , ' . ' ) ;
Fin.

```

9° : Ecrire l'algorithme qui remplit un tableau avec les 10 nombre suivants (Nombres de Fibonacci):
1 , 1 , 2 , 3 , 5 , 8 , 13 , 21 , 34 , 55 .

Corrigé

Familiariser l'étudiant avec les suites et les séries.

Algorithme Fibonacci ;

Var

i : Entier ;

V : Tableau [10] de Entier ;

Début

// Remplissage du tableau //

V [0] := 1 ;

V [1] := 1 ;

Pour i := 2 à 9 faire

V [i] := V [i - 1] + V [i - 2];

// Affichage du tableau //

Pour i := 0 à 8 faire

Ecrire (V [i] , ' , ') ;

Ecrire (V [9] , ' . ') ;

Fin.

10° : Soit un vecteur T. Ecrire un algorithme qui permet de construire deux vecteurs Tp et Tn contenant respectivement les éléments pairs et les éléments impairs du vecteur T.

Corrigé

- Dans cet exercice, on va introduire l'opérateur modulo (**mod**) pour déterminer si un nombre est pair ou impair.
- De plus, on va leur apprendre comment faire un contrôle. C'est-à-dire, comme on a déclaré un tableau avec 30 éléments, l'utilisateur ne doit pas introduire une valeur de N supérieure à 30.

```

Algorithme trois_vects ;
Var
  N , i, j, k : Entier ;
  T, Tp, Tn : Tableau [ 30 ] de Réel ;
Début
  // Contrôle de la valeur de N qui doit être entre 1 et 30 //
  Répéter
    Ecrire ( ' Donnez le nombre des étudiants entre 1 et 30 ' ) ;
    Lire ( N ) ;
  Jusqu'à N >= 1 et N <= 30

  // Remplissage du tableau (ou vecteur) //
  j := -1 ;      // Indice de Tp
  k := -1 ;      // Indice de Tn
  Pour i := 0 à N - 1 faire
    Début
      Ecrire ( ' Donnez la valeur de T [ ' , i , ' ] ' ) ;
      Lire ( T [ i ] ) ;

      // Vérifie si la valeur de T [ i ] est pair ? //
      Si T [ i ] mod 2 = 0 Alors
        Début
          j := j + 1 ;
          Tp [ j ] := T [ i ] ;
        Fin
      Sinon
        // Donc, la valeur de T [ i ] est impair //
        Début
          k := k + 1 ;
          Tn [ k ] := T [ i ] ;
        Fin ;
      Fin ;
    Fin ;

  // Affichage de Tp et Tn //
  Pour i := 0 à j faire
    Ecrire ( ' Tp [ ' , i , ' ] = ' , Tp [ i ] ) ;

  Pour i := 0 à k faire
    Ecrire ( ' Tn [ ' , i , ' ] = ' , Tn [ i ] ) ;

Fin.

```

11° : Ecrire un algorithme qui remplit un vecteur V avec les éléments d'un vecteur T, placés dans l'ordre inverse.

Corrigé

Cet exercice montre comment manipuler l'indice.

```
Algorithme deux_vects ;
Var
  N , i : Entier ;
  T, V : Tableau [ 50 ] de Réel ;
Début
  // Contrôle de la valeur de N qui doit être entre 1 et 50 //
  Répéter
    Ecrire ( ' Donnez le nombre des étudiants entre 1 et 50 ' ) ;
    Lire ( N ) ;
  Jusqu'à N >= 1 et N <= 50

  // Remplissage du tableau (ou vecteur) et inversion du tableau //
  Pour i : = 0 à N - 1 faire
    Début
      Ecrire ( ' Donnez la valeur de T [ ' , i , ' ] ' ) ;
      Lire ( T [ i ] ) ;
      V [ N - 1 - i ] := T [ i ] ;
    Fin ;

  Pour i : = 0 à N - 1 faire
    Ecrire ( ' V [ ' , i , ' ] = ' , V [ i ] ) ;
Fin.
```


12° : Ecrire un algorithme qui remplit un vecteur V puis effectue une permutation circulaire de droite à gauche.

Corrigé

Cet exercice montre comment manipuler l'indice.

Algorithme permutation ;

Var

N , i : Entier ;

V : Tableau [50] de Réel ;

Début

// Contrôle de la valeur de N qui doit être entre 1 et 50 //

Répéter

Ecrire (' Donnez le nombre des étudiants entre 1 et 50 ') ;

Lire (N) ;

Jusqu'à N >= 1 et N <= 50

// Remplissage du tableau (ou vecteur) //

Pour i : = 0 à N - 1 faire

Début

Ecrire (' Donnez la valeur de V [' , i , '] ') ;

Lire (V [i]) ;

Fin ;

// On commence par sauvegarder le 1er élément dans une variable « X » //

X := V [1] ;

// On décale les autres éléments, de 1 à N-1 d'une position vers la droite //

Pour i : = 1 à N - 1 faire

V [i - 1] := V [i] ;

// A la fin, on écrit la valeur de « x », soit « V [1] », dans le dernier élément //

V [N - 1] := X ;

Pour i : = 0 à N - 1 faire

Ecrire (' V [' , i , '] = ' , V [i]) ;

Fin.

Email d'Ahmed-Nacer pour d'éventuels enrichissements, **CORRECTIONS** ou questions.

ahname06@yahoo.fr

ahname6@gmail.com