Reste de la correction des séries

Série Nº2

Exercice 5

1. On doit utiliser le type entier car les termes $n!$, $\frac{n!}{(n-p)!}$ et $\frac{n!}{p!(n-p)!}$ retourne des valeur entière

2. L'algorithme correspondant :

```
Algorithme Dénombrement
Variable n, p, nAp, nCp, nFact, pFatc, npFact, i : Entier
Début
  nAp ← 0
  nCp ← 0
  nFact ← 1
  pFact ← 1
  npFact ← 1
  Répéter
    Ecrire("Donnez le nombre n (n>0) :")
    Lire(n)
  Jusqu'à (n >= 0)
  Répéter
    Ecrire("Donnez le nombre p (p>=0) :")
    Lire(p)
  Jusqu'à (p >= 0)
  Pour i allant de 2 jusqu'à n Faire
    nFact ← nFact*i
  FinPour
  Si n >= p Alors
    Si p = 0 alors
      nAp ← 1
      nCp ← 1
    Sinon
      Pour i allant de 2 jusqu'à p Faire
        pFact ← pFact*i
      FinPour
      Pour i allant de 2 jusqu'à n-p Faire
        npFact ← npFact*i
      FinPour
      nAp ← n_fact/n_p_fact
      nCp ← n_fact/(p_fact*n_p_fact)
    FinSi
  FinSi
  Ecrire("Pn = ", nFact, " nAp = ", nAp, " nCp = ", nCp)
Fin
```

Exercice 6

```
Algorithme Notes
Variable Note, Max, Min : Réel
```

```
          nMax, nMin : Entier
Début
     Max ← -1
     Min ← 21
     nMax ← 0
     nMin ← 0
     Ecrire("Donnez une note (-1 ou 20 pour finir) :")
     Lire(Note)
     TantQue(Note >= 0 Et Note <= 20)
     Si Note = Max Alors
          nMax ← nMax + 1
     FinSi

     Si Note > Max Alors
          Max ← Note
          nMax ← 1
     FinSi
     Si Note = Min Alors
          nMin ← nMin + 1
     FinSi
     Si Note < Min Alors
          Min ← Note
          nMin ← 1
     FinSi
     Ecrire("Donnez une note (-1 ou 20 pour finir) :")
          Lire(Note)
     FinTantQue
     /*si aucune note (c'est-à-dire si max<0) les résultats sont sans signification*/
     Si Max >= 0 Alors
     Ecrire("La note maximale est : ", Max, "Nombre d'occurrence : ", nMax)
     Ecrire("La note minimal est : ", Min, "Nombre d'occurrence : ", nMin)
     FinSi
Fin
```

Série N°3

Exercice 2

```
Algorithme Produit_Scalaire
Variable n, i : Entier
         produit : Réel
Tableau U[20], V[20] : Réel
Début
  Répéter
    Ecrire("Donnez la dimension des vecteurs : ")
    Lire(n)
  Jusqu'à(n > 0 ET n <= 20)
  Ecrire("Saisie du vecteur U : ")
  Pour i allant de 0 Jusqu'à n-1 Faire
    Ecrire("Donnez l'élément ", i+1)
    Lire(U[i])
  FinPour
  Ecrire("Saisie du vecteur V : ")
  Pour i allant de 0 Jusqu'à n-1 Faire
```

```
    Ecrire("Donnez l'élément ", i+1)
    Lire(V[i])
  FinPour
  produit ← U[0]*V[0]
  Pour i allant de 1 Jusqu'à n-1 Faire
    produit ← produit + U[i]*V[i]
  FinPour
  Ecrire("Le produit scalaire est ", produit)
Fin
```

Exercice 3

```
Algorithme Polynôme
Variable i, n : Entier
         X, P_X : Réel
Tableau a[50] : Réel
Début
  Répéter
    Ecrire("Donnez la taille des vecteurs : ")
    Lire(n)
  Jusqu'à(n > 0 ET n <= 50)
  Ecrire("Saisie des valeurs a(i) : ")
  Pour i allant de 0 Jusqu'à n Faire
    Ecrire("Donnez a",i)
    Lire(a[i])
  FinPour
  Ecrire("Donnez la valeur de X : ")
  Lire(X)
  P_X ← a[0]
  Pour i allant de 1 Jusqu'à n-1 Faire
    P_X ← P_X + a[i]*X^i
  FinPour
  Ecrire("P(", X ,") = ", P_X)
Fin
```

Exercice 4 : ***En utilisant une seule matrice A***

```
Algorithme Transposée
Variable n, m, i, j, tmp, max : Entier
Tableau A[50,50] : Entier
Début
  Ecrire("Donnez le nombre de lignes n (0<n<=50) : ")
  Lire(n)
  Ecrire("Donnez le nombre de colonnes m (0<m<=50) : ")
  Lire(m)
  Ecrire("Saisie de la matrice")
  Pour i allant de 0 Jusqu'à n-1 Faire
     Pour j allant de 0 Jusqu'à m-1 Faire
          Ecrire("Donnez l'élément (",i,",",j,") :")
          Lire(A[i,j])
     FinPour
  FinPour
  Ecrire("Affichage de la matrice")
  Pour i allant de 0 Jusqu'à n-1 Faire
     Pour j allant de 0 Jusqu'à m-1 Faire
          Ecrire(A[i,j]," ")
```

```
    FinPour
  FinPour
  Si n > m Alors
    max ← n
  Sinon
    max ← m
  FinSi
  Pour i allant de 0 Jusqu'à max-1 Faire
    Pour j allant de 0 Jusqu'à i-1 Faire
      tmp ← A[i,j]
      A[i,j] ← A[j,i]
      A[j,i] ← tmp
    FinPour
  FinPour
  Ecrire("Affichage de la matrice transposée")
  Pour i allant de 0 Jusqu'à m-1 Faire
    Pour j allant de 0 Jusqu'à n-1 Faire
      Ecrire(A[i,j]," ")
    FinPour
  FinPour
Fin
```

Exercice 5

```
Algorithme Matrice_Produit
Variable N, M, P, i, j, k: Entier
Tableau A[50,50], B[50,50], C[50,50]  : Entier
Début
  Ecrire("Donnez les tailles N, M et P: ")
  Lire(N, M, P)
  Ecrire("Saisie de la matrice A")
  Pour i allant de 0 Jusqu'à N-1 Faire
    Pour j allant de 0 Jusqu'à M-1 Faire
      Ecrire("Donnez l'élément (",i,",",j,") :")
      Lire(A[i,j])
    FinPour
  FinPour
  Ecrire("Saisie de la matrice B")
  Pour i allant de 0 Jusqu'à M-1 Faire
    Pour j allant de 0 Jusqu'à P-1 Faire
      Ecrire("Donnez l'élément (",i,",",j,") :")
      Lire(B[i,j])
    FinPour
  FinPour
  Ecrire("Affichage de la matrice A")
  Pour i allant de 0 Jusqu'à N-1 Faire
    Pour j allant de 0 Jusqu'à M-1 Faire
      Ecrire(A[i,j]," ")
    FinPour
  FinPour
  Ecrire("Affichage de la matrice B")
  Pour i allant de 0 Jusqu'à M-1 Faire
    Pour j allant de 0 Jusqu'à P-1 Faire
      Ecrire(B[i,j]," ")
    FinPour
```

```
  FinPour

  Pour i allant de 0 Jusqu'à N-1 Faire
     Pour j allant de 0 Jusqu'à P-1 Faire
          C[i,j] ← 0
          Pour k allant de 0 Jusqu'à M-1 Faire
               C[i,j] ← C[i,j] + A[i,k]*B[k,j]
          FinPour
     FinPour
  FinPour
  Ecrire("Affichage de la matrice résultat")
  Pour i allant de 0 Jusqu'à N-1 Faire
     Pour j allant de 0 Jusqu'à P-1 Faire
          Ecrire(C[i,j]," ")
     FinPour
  FinPour
```

Exercice 6

5. Matrice_Dynamique

```
Algorithme Matrice_Dynamique
Variable N, M, i, j : Entier
Tableau T : ^^Réel
Début
     Ecrire("Donnez le nombre de ligne et de colonnes :")
     Lire(N, M)
     Allouer(T, N)
     Pour i allant de 0 Jusqu'à N-1 Faire
          Allouer(T[i], M)
     FinPour
     Pour i allant de 0 Jusqu'à N-1 Faire
          Pour j allant de 0 Jusqu'à M-1 Faire
               Ecrire("Donnez l'élément (", i, ",", j, ")")
               Lire(T[i, j])
          FinPour
     FinPour
     Pour i allant de 0 Jusqu'à N-1 Faire
          Pour j allant de 0 Jusqu'à M-1 Faire
               Ecrire(T[i, j], " ")
          FinPour
     FinPour
     Pour i allant de 0 Jusqu'à N-1 Faire
          Libérer(T[i])
     FinPour
     Libérer(T)
Fin
```

Série Nº4

Exercice 4

```
Procédure LireVecteur(T : ^Réel, N : Entier)
Variable i : Entier
Début
     Pour i allant de 0 Jusqu'à N-1 Faire
```

```
          Ecrire("Donnez l'élément ", i+1)
          Lire(T[i])
     FinPour
FinProcédure

Procédure EcrireVecteur(T : ^Réel, N : Entier)
Variable i : Entier
Début
     Pour i allant de 0 Jusqu'à N-1 Faire
          Ecrire(T[i]," ")
     FinPour
FinProcédure

Fonction ProduitVecteur(V1 : ^Réel, V2 : ^Réel, N : Entier) : Réel
Variable i : Entier
         Produit : Réel
Début
     Produit ← V1[0]*V2[0]
     Pour i allant de 1 Jusqu'à N-1 Faire
          Produit ← Produit + V1[i]*V2[i]
     FinPour
     Retourner Produit
FinFonction

Algorithme Test
Variable V1, V2 : ^Réel
         i, N : Entier
         Produit : Réel
Début
     Ecrire("Donnez N :")
     Lire(N)
     LireVecteur(V1, N)
     LireVecteur(V2, N)
     Produit ← ProduitVecteur(V1, V2, N)
     Ecrire("Le produit scalaire est : ", Produit)
Fin
```

Exercice 5

```
Procédure LireMatrice(T : ^^Réel, N : Entier, M : Entier)
Variable i, j : Entier
Début
     Pour i allant de 0 Jusqu'à N-1 Faire
          Pour j allant de 0 Jusqu'à M-1 Faire
               Ecrire("Donnez l'élément ", i+1, ",", j+1, ")")
               Lire(T[i,j])
          FinPour
     FinPour
```

```
FinProcédure

Procédure EcrireMatrice(T : ^^Réel, N : Entier, M : Entier)
Variable i, j : Entier
Début
     Pour i allant de 0 Jusqu'à N-1 Faire
          Pour j allant de 0 Jusqu'à M-1 Faire
               Ecrire(T[i,j], " ")
          FinPour
     FinPour
FinProcédure

Fonction AllouerMatrice(N : Entier, M : Entier) : ^^Réel
Variable i : Entier
         T : ^^Réel
Début
     Allouer(T, N)
     Pour i allant de 0 Jusqu'à N-1 Faire
          Allouer(T[i], M)
     FinPour
     Retourner T
FinFonction

Procédure LibérerMatrice(T : ^^Réel, N : Entier)
Variable i : Entier
Début
     Pour i allant de 0 Jusqu'à N-1 Faire
          Libérer(T[i])
     FinPour
     Libérer(T)
FinProcédure

Fonction TransposéeMatrice(A : ^^Réel, N : Entier, M : Entier) : ^^Réel
Variable i, j : Entier
         tA : ^^Réel
Début
     tA ← AllouerMatrice(M, N)
     Pour i allant de 0 Jusqu'à M-1 Faire
          Pour j allant de 0 Jusqu'à N-1 Faire
               tA[i,j] ← A[j,i]
          FinPour
     Retourner tA
FinFonction

Algorithme Test
Variable A, tA : ^^Réel
        N, M : Entier
```

```
Début
     Ecrire("Donnez N et M :")
     Lire(N,M)
     A ← AllouerMatrice(N, M)
     LireMatrice(A, N, M)
     Ecrire("Affichage de la matrice données :")
     EcrireMatrice(A, N, M)
     tA ← TransposéeMatrice(A, N, M)
     Ecrire("Affichage de la matrice Transposée :")
     EcrireMatrice(tA, M, N)
     LibérerMatrice(A, N)
     LibérerMatrice(tA, M)
Fin
```