

# MODULE: CALCUL FORMEL

## TP N°5

# MAPLE

---

### Exercice 1

Ecrire un programme sous Maple qui permet de concaténer deux listes **L1** et **L2**.

Pour résoudre ce problème suivez les étapes suivantes :

- Définissez une procédure qui ajoute un élément '**a**' à la fin d'une liste **L**, cette procédure est appelée '**add\_elem**' qui prend en arguments une liste **L** et '**a**' et retourne une nouvelle liste à l'aide de la fonction **op**;
- Définissez deux listes **L1** et **L2** respectivement avec les valeurs 2,7,4 et 10,8,23,2;
- Puis dans une boucle '**for**' réalisez la concaténation de **L1** et **L2** en utilisant la procédure '**add\_elem**'.
- Et à la fin, donnez le résultat du programme.

### Exercice 2

Ecrire un programme sous Maple qui permet d'ordonner les éléments d'une liste **L** dans l'ordre décroissant.

Pour résoudre ce problème suivez les étapes suivantes :

- Définissez une procédure appelée « **indice\_max** » qui prend en argument une liste **M**, et retourne le rang (indice dans la liste) de la valeur maximale de cette liste.
- Définissez une liste **L** avec les valeurs 2, 7, 4, 10, 8, 23, 2;
- Puis, dans une boucle, ordonnez les éléments de la liste **L** dans l'ordre **décroissant** en utilisant la procédure « **indice\_max** » ;
- Et à la fin, affichez la liste ordonnée.

**Exemple** : **L** : [4, 5, 3]

L'indice du max est 2

Insérer le max à la fin de la liste résultat **R** : [5]

Supprimer l'élément 5 de la liste **L** : [4, 3]

Répéter ces trois opérations jusqu'à ce que la liste **L** soit vide et **R** : [5, 4, 3]

### Exercice 3

Une matrice creuse est une matrice contenant beaucoup de zéros. Il est ridicule de mémoriser tous les  $m \times n$  coefficients d'une matrice creuse de taille  $(m,n)$ . Il suffit d'en mémoriser les coefficients non nuls.

On caractérise une matrice creuse par 3 listes :

- La liste des indices des lignes ;
- La liste des indices des colonnes ;
- La liste des valeurs.

Par exemple la matrice suivante :

$$\begin{bmatrix} 1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 3 & 9 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 4 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

est représentée par :

Valeurs = [ 1, 2, 1, 3, 9, 1, 4 ]

Lignes = [ 1, 1, 1, 2, 2, 3, 3 ]

Colonnes = [ 1, 2, 9, 2, 3, 2, 3 ]

On suppose que la matrice est de dimension (N; M) et que les trois listes (valeurs, lignes et colonnes) sont déjà remplies.

En utilisant Maple :

- 1) Définissez une fonction **Get\_value** qui prend en argument les indices i et j d'une matrice et renvoie la valeur de cette case.
- 2) Définissez une procédure appelée «**Get\_ligne**» qui prend en argument l'indice d'une ligne et renvoie les éléments de cette ligne en utilisant la fonction **Get\_value**.

## Exercice 4

Les informations d'une série statistique quantitative discrète sont représentées sous forme de deux listes. Les éléments de la première liste (**L1**) sont les valeurs observées  $X_i$  et les éléments de la deuxième liste (**L2**) sont les effectifs  $n_i$  (Fréquences d'apparition des valeurs observées).

L1 = [10, 20, 30, 40, 50]

L2 = [3, 2, 2, 4, 1]

1. Ecrire un programme sous MAPLE qui permet de calculer la moyenne arithmétique ainsi que la variance de la série statistique représentée par **L1** et **L2**.
2. Ecrire un programme pour calculer la valeur la plus fréquente dans cette série statistique (appelée valeur modale).

**Nb :**

$$\bar{X} = \frac{1}{n} \cdot \sum_{i=1}^n n_i x_i$$

$$V = \frac{1}{n} \sum_{i=1}^n (n_i x_i - \bar{X})^2$$

tel que :

$n$  est le nombre d'éléments de la série.

$\bar{X}$  est la moyenne de la série statistique à une variable quantitative discrète de taille  $n$ .

$x_i$  sont les valeurs observées, ou les éléments de la série statistique.

$n_i$  sont les effectifs des valeurs observées.

$V$  est la variance.

## Exercice 5

Soit une fonction définie comme suit :  $f(x) = e^{\sin(1/x^2)}$ .

1. Définissez la fonction  $f$  sous MAPLE ;
2. Construisez la liste LF des  $f(x_i)$  pour  $i = 1..10$  à l'aide de la commande seq().
3. Définissez une procédure « **distEuclid** » qui prend en argument une liste et renvoie la distance euclidienne entre le premier et le dernier élément de cette liste.

4. Appeler cette procédure en lui donnant en argument la liste LF de la deuxième question.

**Rappel** : La distance euclidienne entre deux points dans un espace à deux dimensions est calculée par la formule :  $d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$

# CORRECTION

---

## Exercice 1

```
add_elem := proc(L, a)
    [op(L), a];
end:
L1 := [2, 7, 4]:
L2 := [10, 8, 23, 2]:
for i from 1 to nops(L2) do
    L1 := add_elem(L1, op(i, L2));
od:
L1;
```

Le résultat [2,7,4,10,8,23,2]

## Exercice 2

```
maximum:=proc(L)
    local rang,max1, i;
    rang:=1:
    max1:=op(1,L):
    for i from 2 to nops(L) do
        if op(i,L)>max1 then
            max1:=op(i,L);
            rang:=i;
        fi:
    od;
    rang;
end:
L:=[2, 7, 4, 10, 8, 23, 2]:
M:=L;
R:=[];
for i from 1 to nops(L) do
    s:=maximum(M):
    R:=[op(R), op(s,M)];
    M:=[seq(op(n,M), n=1..s-1), seq(op(n,M), n=s+1..nops(M))];
od:
R;
end:
```

## Exercice 3

```
lignes:=[1, 1, 1, 2, 2, 3, 3 ]:
```

```

colonnes:=[1, 2 ,9, 2 ,3 ,2, 3 ]:
valeurs := [1, 2, 1, 3, 9 ,1 ,4 ]:
N:=3:
M:=9:

get_value:=proc(i, j)
local k,l,resultat:
resultat:=0:
for k from 1 to nops(colonnes) do
if( op(k,colonnes)=j and op(k,lignes)=i ) then
resultat:=op(k,valeurs);
fi:
od:
resultat;
end:

get_ligne :=proc(i)
local L, j:
L:=[]:
for j from 1 to M do
L:=[op(L), get_value(i, j)]:
od:
L;
end:

get_ligne(3);
get_value(3,3);

```

## Exercice 4

**Calcul de la moyenne arithmétique :**

```

L1:=[10, 20, 30, 40, 50]:
L2:=[3, 2, 2, 4, 1]:
MoyenneArithmetique:=0:
sommeL2:=0:
for c from 1 to nops(L1) do
MoyenneArithmetique:=MoyenneArithmetique+op(c, L1)*op(c, L2):
sommeL2:=sommeL2+op(c, L2):
od:
MoyenneArithmetique:=MoyenneArithmetique/sommeL2:
evalf(MoyenneArithmetique);

```

**Calcul de la variance :**

```

Variance:=0:
for c from 1 to nops(L1) do
Variance:=Variance+op(c, L2) * (op(c, L1) -MoyenneArithmetique) ^2:
od:
Variance:=Variance/sommeL2:
evalf(Variance);

```

**Ou bien, deuxième solution**

```

Variance:=0:
for c from 1 to nops(L1) do
Variance:=Variance+op(c,L2)*(op(c,L1))^2:
od:
Variance:=Variance/sommeL2- (MoyenneArithmetique)^2:
evalf(Variance);

```

**Calcul de la valeur modale :**

```

valmax:=0:
for c from 1 to nops(L2) do
if (valmax<op(c,L2)) then
valmax:=op(c,L2):
ind:=c:
fi:
od:
op(ind,L1);

```

## Exercice 5

```

f:=(x)->exp(sin(1/x**2)):
LF:=[seq(f(x),x=1..10)];

```

```

distEuclid:=proc(L)
local x1,x2,y1,y2;
x1:=1:y1:=L[x1]:
x2:=nops(L):y2:=L[x2]:
evalf(sqrt((x1-x2)**2 + (y1-y2)**2));
end:

```

```

distEuclid(LF);

```