

Listes et matrices

Bonjour à tous, ce nouveau cours concerne la mémoire de votre calculatrice. En effet vous vous êtes sans doute rendu compte que l'on est vite limité avec 26 variables. Comment stocker plus d'informations, faire des sauvegardes dans les jeux ou trier un grand nombre de valeurs numériques ?

Nous avons à disposition 2 structures :

- Les listes, un élément est repéré par sa position dans la liste.

	List 1	List 2	List 3	List 4
SUB				
1	0			
2	0			
3	0			
4	0			

GRAPH CALC TEST DATA DIST

- Les matrices, ce sont des tableaux à 2 dimensions, un élément est repéré par sa ligne puis sa colonne.

A	1	2	3	4 →
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0
4	0	0	0	0
5	0	0	0	0

ROP ROW COL EDIT

Nous allons donc parler des listes et des matrices, leur utilisation et leur syntaxe sont très proches.

Les listes

Il y a 26 listes disponibles, avec un maximum de 999 cases chacune (255 pour les anciens modèles comme la Graph 35/65/80/100 (+)). Pour assigner (mettre) une valeur dans la liste, on utilise la syntaxe :

7 → List2[5] // on met 7 dans la 5^{ème} case de la liste 2

N → List1[X] // ça marche aussi avec des variables sauf pour le numéro de la liste, ici on met la valeur de N dans la X^{ème} case de la liste 1

On obtient la commande List avec [SHIFT] [1].

Pour récupérer une valeur on fait :

List2[5] → N // on met la valeur de la 5^{ème} case de la liste 2 dans la variable N

Cependant si vous testez ces exemples, ils ont de fortes chances de ne pas marcher, en effet avant d'assigner une valeur dans une liste il faut au préalable « créer » la liste. Pour cela on utilise la commande Dim que vous trouverez dans le menu liste en faisant [OPTN] [F1] (List) [F3] (Dim).

```
5 → Dim List 1 // et voilà une liste de 5 cases à été créée
```

Une petite astuce à connaître et qui est bien pratique est que la première case d'une liste vide, et la case qui suit immédiatement une case créée, sont automatiquement créées lors de l'assignation.

Vous pouvez aussi créer et remplir simultanément une liste en faisant :

```
{1, 2, 3} → List 1 // crée la liste 1 et la remplit avec dans l'ordre 1, 2, 3
```

Vous maîtrisez désormais tout sur les listes, comment les créer, leur assigner des valeurs et les récupérer.

Les matrices

C'est tout aussi simple pour les matrices. Par contre les matrices sont désignées par des lettres allant de A à Z (il y a donc aussi 26 matrices). La commande Mat s'obtient avec [SHIFT] [2].

Les matrices doivent être obligatoirement créées, on utilise la syntaxe suivante :

```
{5,8} → Dim Mat A // cela crée une matrice de 5 lignes et 8 colonnes
```

Vous pouvez ensuite mettre et récupérer des valeurs dans votre matrice :

```
N → Mat A[X, Y] // on met la valeur N dans la case de la matrice A sur la Xème ligne et Yème colonne, on peut évidemment utiliser des variables  
Mat A[X, Y] → N // la variable N prend la valeur de la case de coordonnées (X, Y)
```



Attention !

Prenez garde à rester dans les dimensions de la matrice lorsque vous utilisez des variables. Par exemple pour reprendre notre exemple X doit être compris entre 1 et 5 et Y entre 1 et 8.

De même que pour les listes, vous pouvez créer et remplir simultanément une matrice en faisant :

```
[[1, 2][3, 4]] → Mat A // crée la matrice A comme cela :  $\begin{matrix} 1 & 2 \\ 3 & 4 \end{matrix}$ 
```

Petites propriétés utiles

On peut transférer entièrement le contenu d'une liste ou d'une matrice dans une autre en faisant :

```
List 2 → List 3 // pour les listes  
Mat A → Mat C // pour les matrices
```

Accéder au contenu des listes et des matrices manuellement :

- Pour les listes il suffit d'aller dans le menu Stat de votre calculatrice.
- Pour les matrices, vous les trouverez dans le menu Run, puis en faisant [F3] (>Mat).

Entrainement

Pour vous entrainer, saurez-vous me dire ce que fait le programme suivant (la réponse est cachée dans ce cours).

```
{3, 3} → Dim Mat A  
  
For 1 → I To 3  
For 1 → J To 3  
  Int (20Ran#) → Mat A[I, J]  
Next: Next  
  
9 → Dim List 1  
For 1 → L To 9  
0 → M: 0 → X: 0 → Y  
  For 1 → I To 3  
  For 1 → J o 3  
    If Mat A[I, J] > M  
    Then Mat A[I, J] → M  
    I → X: J → Y  
  IfEnd  
  Next: Next  
M → List 1[L]  
0 → Mat A[X, Y]  
Next
```