TP 9 – Listes et matrices- Introduction

Les listes et la matrcies permettent de stocker plusieurs valeurs dans une même variable. Nous allons voir comment les créer et les utiliser.

I Les listes

I.1 Créer une liste

```
liste = [ a1, a2 , ... , an ]
```

où `a1`, `a2`,…, `an` sont les valeurs à stocker dans la liste.

Exemple.

```
L1 = [3 , 2.1, 5]
```

I.2 Liste d'entiers consécutifs

```
L = [ n : m ]
```

Exemple. Définir les listes L2 = [0,1,2,3,4] et L3 = [3,4,5,6,7,8]

```

```

I.3 Manipulations de listes

On considère une liste appelée `L`.

Opération	Syntaxe Scilab
Élément numéro k de L Modifier l'élément numéro k	`L(k)` `L(k) = valeur`
Longueur de L	`length(L)`
Ajouter un élément x à la fin de L	`L = [ L , x ]`

Exemple.

1. Définir la liste $L3 = [4, 7, 8, 2, 1, 0, 6]$
2. Changer le troisième élément (le 8 donc) en un 10.
3. Ajouter un 4 à la fin de $L3$.

```
-->

-->

-->
```

Exemple. Définir, à l'aide d'une boucle, la liste $L = [1, 2, 4, 8, 16, \ldots, 2^{20}]$.

1. Méthode 1 : on part d'une liste videet on ajoute un à un les éléments/

```
L = []  // liste vide
for k = ... : ...


end
disp(L)
```

2. Méthode 2 : on part d'une liste de taille voulue et on modifie un à un les éléments.

```
L = [ 1 : ...] // liste à la bonne taille
// on pourrait aussi partir d'une liste composée de 0
// (voir plus loin)
for k = ... : ...


end
disp(L)
```

II Les matrices et tableaux de valeurs

Avec Scilab, nous pouvons facilement créer des matrices qui permettent de stocker plusieurs valeurs sous la forme d'un tableau rectangulaire. Remarquons que les listes sont en fait des matrices à une ligne, donc ce que nous allons voir ici permet également de compléter la partie sur les listes.

II.1 Créer une matrice

La matrice $A = \begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1p} \\ a_{21} & a_{22} & \ldots & a_{2p} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \ldots & a_{np} \end{pmatrix}$ peut être définie ainsi :

```
A = [ a11, a12 , ... , a1p ; a21, a22 , ... , a2p ; ... ; an1, an2 , ... , anp ]
```

Exemple. Définir la matrice $A = \begin{pmatrix} 1 & 4 \\ 2 & 7 \\ -1 & 5 \end{pmatrix}$

```

```

II.2 Matrices particulières

- Matrice nulle de taille $n \times p$: `zeros(n,p)`
- Matrice identité de taille $n \times n$: `eye(n,n)`
- Matrice aléatoire de taille $n \times p$: `rand(n,p)`
  Les éléments de cette matrice sont des nombres aléatoire de $[0, 1]$.

II.3 Manipulation de matrices

On considère une matrice appelée `M`.

Opération	Syntaxe Scilab
Coefficient (i, j) de M Modifier le coefficient (i, j)	`M(i,j)` `M(i,j) = valeur`
Ligne i de M Colonne j de M	`M(i,:)` `M(:, j)`
Taille de M	`n , p = size(M)`
Rassembler deux matrices	`[M1 , M2]` (côte à côte) `[M1 ; M2]` (l'une en dessous de l'autre)

Exemple.

1. Définir les matrices $M1 = \begin{pmatrix} 1 & 2 & 3 \\ 8 & 5 & 6 \end{pmatrix}$ et $M2 = \begin{pmatrix} 7 & 1 \\ -1 & 4 \end{pmatrix}$
2. Changer le 8 de la matrice $M1$ en un 4.
3. Rassembler $M1$ et $M2$ en une seule matrice, appelée $M3$.
4. Afficher la deuxième ligne de $M1$.

```
--> M1 =

--> M2 =

-->

-->

-->
```

II.4 Calculs sur les matrices

Les opérations classiques `+`, `*`, `-`, `^` sont pour Scilab des **opérations matricielles**.

a) Opérations matricielles (= opérations normales)

Action	Syntaxe Scilab
Somme	`A + B`
Produit matriciel	`A * B`
Puissance matricielle	`A ^ n`
Inverse	`inv(A)`
Tranposée	`A'`

b) Opérations terme à terme : .* ./ .^

On va aussi avoir besoin de réaliser des opérations terme à terme sur des listes ou tableaux de valeurs.

Soient `T1 = [a11, a12, ..., anp]` et `T2 [b11, b12, ..., bnp]` deux tableaux (ou listes) de même taille.

Remarque : pour les combinaisons linéaires, l'opération classique est déjà terme à terme. Nous les rappelons quand même ici.

Opération	Syntaxe Scilab	Résultat
Addition terme à terme	`T1 + T2`	`[ a11 + b11, ..., anp + bnp ]`
Ajout d'un nombre	`T1 + c`	`[ a11 + c, ..., anp + c`
Multiplication par un nombre	`c * T1`	`[ c * a11, ..., c * anp ]`
Multiplication terme à terme	`T1 .* T2`	`[a11 * b11, ..., anp * bnp ]`
Division terme à terme	`T1 ./ T2`	`[ a11 / b11, ..., anp / bpn ]`

Puissance terme à terme	`T1 .^ k`
	`k .^ T1`
	`T1 .^ T2`
		`[ a11^k, ..., anp^k ]`
		`[ k^a11, ..., k^anp ]`
		`[ a11^b11, ..., anp^bnp ]`

Fonctions usuelles	`exp(T1), log(T1), ...`	`[ exp(a11), ..., exp(anp) ]`
Somme des termes Produit des termes	`sum(T1)` `prod(T1)`	

Application directe : calculer une somme / un produit en une ligne

Exemple.

1. Définir la liste $[1^3, 2^3, 3^3, \ldots, 100^3]$.
2. En déduire, en une ligne $S = \sum_{k=1}^{100} k^3$.

```
-->

--> S =
```

Exemple. Définir en une ligne $n! = \prod_{k=1}^{n} k$ (on suppose que n est donné).

```
-->
```

Exercice 1

1. Définir les listes $L_1 = [1, 2, 3, \dots, 10]$ et L_2 une liste de 10 nombres aléatoires de $[0, 1]$.

```
L1 =

L2 =
```

2. Calculer la liste obtenue en faisant le produit terme à terme de ces deux listes.

3. Modifier l'élément numéro 3 dans $L2$ pous qu'il soit égal à 7.

4. Ajouter un 11 à la fin de L_1.

5. Définir les matrices $A = \begin{pmatrix} 1 & 2 & 3 \\ 5 & 4 & -1 \end{pmatrix}$ et $B = \begin{pmatrix} 0 & 1 & 2 \\ -3 & 2 & 0 \\ 1 & 2 & 4 \end{pmatrix}$

```
A =

B =
```

6. Calculer le produit matriciel AB.

7. Afficher la tranposée de A.

8. Calculer l'inverse de B (si elle n'est pas inversible, il y aura un message d'erreur).

9. Calculer B^3 (puissance matricielle).

10. Définir $C = \begin{pmatrix} 0^3 & 1^3 & 2^3 \\ (-3)^3 & 2^3 & 0^3 \\ 1^3 & 2^3 & {}^{3}4 \end{pmatrix}$ à partir de B.

11. Modifier la ligne 2 de A en $[7, 8, 9]$.

12. Créer rapidement la matrice $D = \begin{pmatrix} 1 & 2 & 3 \\ 7 & 8 & 9 \\ 0 & 1 & 2 \\ -3 & 2 & 0 \\ 1 & 2 & 4 \end{pmatrix}$ à partir de A et B.

Exercice 2

1. Écrire la liste $[1, 2, \dots, 1000]$ puis la liste $[1^{-2}, 2^{-2}, \dots, 1000^{-2}]$.

2. En déduire une commande (en une ligne) pour calculer $\displaystyle\sum_{k=1}^{1000} \frac{1}{k^2}$.

3. Vérifier que $\displaystyle\lim_{n \to +\infty} \sum_{k=1}^{n} \frac{1}{k^2} = \frac{\pi^2}{6}$.

Exercice 3

1. Définir la matrice $M = \begin{pmatrix} 1 & 3 & -2 \\ 0 & 4 & -3 \\ -2 & 1 & 0 \end{pmatrix}$ puis calculer $M^3 - 5M^2 + 3M$.

```

```

2. En déduire, à la main, une expression de l'inverse de M en fonction de M^2, M et I_3.

3. À l'aide de Scilab, trouver l'expression de cette matrice, qui sera ici notée N (sans utiliser `inv`).

```
N =
```

4. Vérifier que N est bien l'inverse de M :
   (a) en calculant le produit MN ;
   (b) en utilisant la commande `inv(M)`.

Exercice 4

1. On souhaite définir, avec des boucles for, la matrice $E = (e_{i,j})_{\substack{1 \leqslant i \leqslant 5 \\ 1 \leqslant j \leqslant 6}}$ avec $e_{i,j} = i^j$

```
E = zeros(... , ...)  // matrice nulle de même taille que E

for i = 1 : ...
    for j = 1 : ...
        E(i,j) = ....   // on modifie l'élément numéro (i,j)
    end
end

disp(E, 'E=')
```

2. De même, définir la matrice $H = (h_{i,j})_{\substack{1 \leqslant i \leqslant 10 \\ 1 \leqslant j \leqslant 4}}$ avec $h_{i,j} = (i+2j)^2$.

```

```

Exercice 5 On considère la suite (u_n) définie par $u_0 = 3$ et, pour tout $n \in \mathbb{N}$, $u_{n+1} = \sqrt{nu_n + 2}$.
Écrire un programme qui demande de saisir un entier n puis crée la liste $L = [u_0, u_1, \ldots, u_n]$

```
n =

u =

L = [u]  // on met le u0 dans L

for k = ... : ...

    u =

    L =

end

disp(L, 'L=')
```

Exercice 6 On considère la suite donnée par

$$u_0 = 0, \quad u_1 = 1 \quad \text{et} \quad \forall n \in \mathbb{N}, \; u_{n+2} = u_{n+1} + u_n.$$

Écrire une fonction qui, étant donné un entier naturel n, renvoie la liste $[u_0, u_1, \ldots, u_n]$ des $n+1$ premiers termes de cette suite.