

Feuille de TP N°04 – Scripts dans Matlab

Le but de ce TP est d'écrire des scripts et fonctions Matlab pour résoudre des problèmes mathématiques avancés.

Exercice 01 :

Ecrire un script Matlab qui demande à l'utilisateur de saisir un nombre entier positif N , puis calcule et affiche la somme des chiffres qui composent ce dernier.

Indications à donner aux étudiants : expliquer la fonction *mod* qui calcule le reste de division, et la fonction *floor* qui calcule l'arrondi vers l'entier inférieur d'un nombre réel.

```
a = input('Veuillez saisir un nombre entier S.V.P : ');
s = 0;
while a ~= 0
    s = s + mod(a, 10);
    a = floor(a / 10);
end;
display(s);
```

Exercice 02 :

1. Ecrire une fonction Matlab qui prend comme argument un vecteur V et une valeur numérique x et qui retourne le nombre d'occurrences de x dans V .

Solution 1

```
function oc = nbreOccurrences(x, v)
    oc = 0;
    for i = 1:length(v)
        if v(i) == x
            oc = oc + 1;
        end;
    end;
```

Solution 2

```
function oc = nbreOccurrences(x, v)
    oc = 0;
    for i = v
        if i == x
            oc = oc + 1;
        end;
    end;
```

Solution 3

```
function oc = nbreOccurences(x, v)
    oc = length(v(v == x))
```

2. Tester votre fonction à partir de la fenêtre de commandes.

Simplement écrire une commande qui fait appel à la fonction précédente, exemple :

```
>> nbreOccurences(5, [1 2 3 4 5 6 5 4])
```

Exercice 03 :

Ecrire un script Matlab qui demande à l'utilisateur de saisir un nombre entier positif n , et puis calcule et affiche les n premiers termes de la suite U_n définie comme suit.

$$U_1 = 2,$$

$$U_2 = 3,$$

$$U_i = \frac{2}{3}U_{i-1} - \frac{1}{4}U_{i-2}$$

```
n = input('Veuillez saisir le nombre de termes : ');
U(1) = 2;
display(U(1));
U(2) = 3;
display(U(2));
for i = 3:n
    U(i) = (2/3) * U(i-1) - (1/4) * U(i-2);
    display(U(i));
end;
```

Exercice 04 :

1. Ecrire une fonction Matlab qui permet de comparer deux vecteurs, puis retourner 1 s'ils sont identiques et 0 sinon.

```
function res = identiques(v1, v2)
    res = 1;
    if length(v1) == length(v2)
        for i = 1:length(v1)
            if v1(i) ~= v2(i)
                res = 0;
            end;
        end;
    else
        res = 0;
    end;
```

2. Ecrire une fonction Matlab qui permet de supprimer toutes les colonnes identiques d'une matrice donnée comme argument (en cas d'existence de deux colonnes identiques il faut garder une seule).

```
function M2 = supIdentiques(M1)
    M2(:,1) = M1(:,1);
    for i = 2:size(M1, 2)
        existe = 0;
        for j = 1:size(M2, 2)
            if identiques(M1(:,i), M2(:,j))
                existe = 1;
            end;
        end;
        if ~existe
            M2 = [M2 M1(:,i)];
        end
    end;
end;
```

3. Ecrire un script Matlab qui permet à l'utilisateur de saisir une matrice, supprime toutes les colonnes identiques puis affiche le résultat.
 - Le script demande à l'utilisateur d'abord de saisir les dimensions de la matrice.
 - L'utilisateur saisit les éléments de la matrice : un élément à la fois.
 - Finalement les colonnes identiques doivent être supprimées et la matrice est affichée après suppression des colonnes identiques.

```
n = input('donnez le nombre de lignes : ');
m = input('donnez le nombre de colonnes : ');
for i = 1:n
    for j = 1:m
        M1(i,j) = input('saisir un element : ');
    end;
end;
M2 = supIdentiques(M1);
display(M2);
```

Exercice 05 :

1. Ecrire une fonction Matlab qui prend comme argument d'entrée un entier positif N . et qui retourne comme résultat un tableau contenant tous les diviseurs de N .

Exemple : Entrée $\rightarrow 9$; sortie $\rightarrow [1, 3, 9]$.

Entrée $\rightarrow 10$; sortie $\rightarrow [1, 2, 5, 10]$.

Entrée $\rightarrow 11$; sortie $\rightarrow [1, 11]$.

```
function v = diviseurs(n)
    v = [];
    for i = 1:n
        if mod(n,i) == 0
            v = [v i];
        end;
    end;
end;
```

2. Ecrire une fonction Matlab qui prend comme argument d'entrée un entier positif N . et qui détermine si N est un nombre premier ou pas. La fonction doit retourner 1 si le nombre est premier et 0 sinon.

Solution 1

```
function p = premier(n)
    if prod(diviseurs(n)) == n
        p = 1;
    else
        p = 0;
    end;
end;
```

Solution 2

```
function p = premier(n)
    if length(diviseurs(n)) <= 2
        p = 1;
    else
        p = 0;
    end;
end;
```

3. Ecrire un Script Matlab qui demande à l'utilisateur de saisir un nombre entier positif N , puis si le nombre est premier affiche : « le nombre (valeur de N) est premier ». Sinon afficher « le nombre (valeur de N) n'est pas premier ».

```
n = input('Veuillez saisir un nombre : ');
if premier(n)
    display(strcat('le nombre '), num2str(n), ' est premier');
else
    display(strcat('le nombre '), num2str(n), ' n''est pas premier');
end;
```

4. Ecrire une fonction qui permet de dire si un nombre est parfait ou pas. Un nombre est parfait s'il est égal à la somme de ses diviseurs (sans compter le nombre lui-même). La fonction doit retourner 1 si le nombre est parfait et 0 sinon.

```
function p = parfait(n)
    if sum(diviseurs(n)) - n == n
        p = 1;
    else
        p = 0;
    end;
```

5. Ecrire un Script Matlab qui demande à l'utilisateur de saisir un nombre entier positif N , puis si le nombre est parfait affiche : « le nombre (valeur de N) est parfait ». Sinon afficher « le nombre (valeur de N) n'est pas parfait ».

```
n = input('Veuillez saisir un nombre : ');
if parfait(n)
    display(strcat('le nombre '), num2str(n), ' est
parfait');
else
    display(strcat('le nombre '), num2str(n), ' n''est pas
parfait');
end;
```