

TP 1 - Matlab

Manip. 1

Créer quatre matrices A , B et C , et D de taille 4×4 de votre choix.

- Former la matrice X de taille 8×8 ayant la forme suivante :

$$X = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$$

- Créer la matrice Y de taille 8×8 ayant la forme suivante :

$$Y = \begin{bmatrix} A & 0 \\ 0 & A \end{bmatrix}$$

- Former les vecteurs $d1$, $d2$, $d3$ et $d4$ à partir des éléments des diagonales des matrices respectivement A , B , C et D .
- Calculer les valeurs maximales de ces vecteurs.
- Former un vecteur-colonne d à partir des vecteurs $d1$, $d2$, $d3$ et $d4$.
- A partir des matrices A , B , C et D former une matrice Z de taille $4 \times 4 \times 4$ (cube)

Manip. 2

Soient les fonctions suivantes pour x compris entre -10 et 10 :

$$F(x) = 0.5/(x + 0.5e^{(x/10)}) , G(x) = 2x / (1 - x^3) , H(x) = 1 / (1 - x^2) \text{ et } I(x) = 10x \sin(1/x)$$

- Tracer les courbes des fonctions dans un même graphe avec des couleurs différentes
- Tracer les courbes des fonctions dans des graphes séparés dans un canevas partagé en deux lignes et deux colonnes
- Donner des titres pour chaque graphe.

TP 2 - Matlab

Manip. 1

Ecrire un programme qui permet de lire une ligne de texte à partir du clavier et la stocker dans une variable L . On souhaite par la suite crypter cette ligne avec un code que l'on doit saisir dans une autre variable C.

- Affecter la ligne codée à une nouvelle chaîne LC.
- Afficher le contenu du LC.
- Décrypter la chaîne LC puis l'affecter à une nouvelle chaîne LD
- Comparer L et LD

Réponse

```
L = input('saisir une ligne de texte', 's');
```

```
C = input('saisir un code pour crypter la chaîne L', 's');
```

```
LC = char((M + C));
```

```
disp(LC)
```

```
LD = char(abs(LC) - C);
```

```
strcmp(L, LD)
```

TP 3 - Matlab

Manip. 1

Ecrire un programme qui calcule et affiche les solutions de l'équation de second degré dans R puis dans C. (les coefficients de l'équation devront être saisis à partir du clavier).

N.B. Vérifier le résultat du programme en utilisant la fonction de matlab :

$[x1, x2] = \text{roots}([a \ b \ c])$; (elle permet de calculer les racines d'un polynôme de degré 2 avec les coefficients a, b et c) ;

Manip. 2

Ecrire un programme qui permet de saisir les éléments d'une matrice A carrés 4x4 à partir du clavier. Construire une nouvelle matrice B de même taille que A selon la condition :

$$B_{ij} = A_{ij} * A_{ij} \quad \text{si } A_{ij} < 0$$

$$B_{ij} = A_{ij} / 2 \quad \text{sinon}$$

- Calculer et afficher les produits des éléments de chaque colonne de B.
- Calculer et afficher la moyenne des éléments de la diagonale de B.

N.B. Vérifier le résultat du programme en utilisant les fonctions de matlab *prod* , *diag* et *mean*

Réponse :

```
for i=1:4
    for j=1:4
        A(i,j)=input('donner une valeur');
        if A(i,j)<0
            B(i,j)=A(i,j)*A(i,j);
        else
            B(i,j)=A(i,j)/2;
        end
    end
end
end
```

```

for j=1:4
    P(j)=1;
    for i=1:4
        P(j)=P(j)*B(i,j);
    end
end
disp(P);

M=0;
for i=1:4
    M=M+B(i,i)/4;
end
disp(M);

```

Manip. 3

Ecrire un programme qui permet de stocker une ligne de texte, saisie à partir du clavier, dans un tableau L. Puis envoyer le texte inversé de L vers un autre tableau R.

Afficher le nouveau texte inversé de R.

Exemple :

L = ' je lis un texte ', → R = ' etxet nu sil ej'

N.B. Ecrire le même programme avec deux boucles différentes (for...end et while ...end)

Réponse :

```

L=input('saisir une ligne de texte:');
N=length(L);
for i=1:N
    R(i)=L(N-i+1);
end
disp(R);

```

TP 4 - Matlab

Manip.1

Ecrire une fonction qui reçoit en entrée une chaîne de caractère et un symbole, et renvoi en sortie le nombre d'occurrence de ce symbole dans la chaîne.

Réponse

```
function N = nbr_symbole (ch, s)
L=length(ch) ;
N=0 ;
for i=1 : L
if ch(i) == s
N=N+1 ;
end
end
```

Manip.2

Ecrire une fonction qui calcule les $n^{\text{ème}}$ termes (n passé en argument) de la suite de Fibonacci :

$$U_0 = 1 \quad U_1 = 1$$
$$U_{n+1} = U_n + U_{n-1}$$

Réponse :

```
function U = fib(n)
if n==0 || n==1
    U=1;
else
    U=fib(n-1)+fib(n-2);
end
```

Manip.3

a)

Ecrire une fonction qui calcule les $n^{\text{ème}}$ termes (n passé en argument) des suites entières U_n et V_n définies ci-dessous et qui les retourne sous forme d'un tableau (la première ligne contiendra les V_n et la seconde ligne les U_n).

$$\begin{array}{ll} U_0 = 1 & V_0 = 1 \\ U_n = V_{n-1} + 1 & V_n = 2U_n - 1 \end{array}$$

b)

Ecrire une fonction `select_terme` qui permet de sélectionner le $n^{\text{ème}}$ terme d'une des suites, en fonction d'un des arguments d'entrée, `choix` (si `choix = 1`, on donnera en sortie U_n ; si `choix = 2`, on donnera en sortie V_n ; sinon, la sortie donnera les deux: $[U_n; V_n]$). Utilisez la fonction définie précédemment et l'instruction `switch...case`.

Réponse :

```
function U=Un(n)
if n==0
    U=1;
else
    U= Vn(n-1)+1;
end
```

```
function V=Vn(n)
if n==0
    V=1;
else
    V= 2*Un(n)-1;
end
```

```
function A=select_terme(n,choix)
switch(choix)
    case 1
        A=Un(n);

    case 2
        A=Vn(n);

    otherwise
        A=[Un(n) Vn(n)];
end
```

TP 5 - Matlab

Manip 1

1. Ecrire deux fonctions *codch* et *decodch* qui reçoivent comme paramètres d'entrée une ligne de texte et une variable numérique. La fonction *codch* renvoie en sortie une ligne de texte codée avec le code contenu dans la variable numérique, tandis que la fonction *decodch* permet le décodage (opération inverse).
(Le codage de la ligne consiste par exemple à faire une opération arithmétique entre la valeur de la variable et le code ascii de la ligne du texte).
2. Créer trois fichiers texte (fich1.txt, fich2.txt et fich3.txt) ;
3. Ouvrir le fichier fich1.txt avec la commande : *edit* fich1.txt et saisissez quelques lignes de texte et sauvegarder le contenu.
4. Ouvrir le fichier fich1.txt en mode lecture et le fichier fich2.txt en mode écriture (ajout).
Utiliser la fonction *codch* pour coder chaque ligne de fich1.txt et envoyer le résultat (texte codé) dans le fich2.txt. (Utiliser la fonction *fgets* ou *fgetl* pour lire les lignes de texte du fichier)
5. Vérifier le contenu du fich2.txt avec la commande *type* fich2.txt
6. Utiliser la fonction *decodch* pour décoder le contenu du fichier fich2.txt. le résultat de décodage sera enregistré dans le fichier fich3.txt

Réponse

1.

```
function B=codch(A,n)
```

```
A=double(A);
```

```
B=A+n;
```

```
B=char(B);
```

```
function B=decodch(A,n)
```

```
B=double(A)-n;
```

```
B=char(B);
```

2.

```
a=fopen('fich1.txt','w');
```

```
b=fopen('fich2.txt','w');
```

```
c=fopen('fich2.txt','w');
```

```
// lecture et codage des lignes de fich1.txt
```

```
m=fopen('fich1.txt','r');
```

```
n=10;
```

```
i=1;
```

```
while (i>0)
```

```
    i=fgets(m);
```

```
    B=codch(i,n);
```

```
    fprintf(b,'%s\n',B);
```

```
end
```

```
z=fopen('fich2.txt','r');
```

```
x=fopen('fich3.txt','a');
```

```
// lecture des lignes de fich2.txt et décodage dans fich3.txt
```

```
j=1;
```

```
while (j>0)
```

```
    j=fgets(z);
```

```
    B=decodch(j,n);
```

```
    fprintf(x,'%s\n',B);
```

```
end
```