Université A. MIRA de Béjaïa-Faculté de la Technologie
Département de Technologie - 2ème année

Examen Final de MATLAB (LCS)

Jan 2014
(02 heures)

Exercice 01: (05 points)

Qu'obtient-on lorsqu'on exécute les instructions suivantes :

```
>> A=[2 : -2: -3 ; -1: 3: 6]
>> B=[1 : 1: 2; 0: 3: 4]
>> C=[diag(2*eye(3)-2,1); 3; 5]
>> D=A.*[A'+1]'-[B , ones(2,1)]
>> E=B.^ones(2) + B .^zeros(2)
>> F=C(end :-2 :1, : )
>> G=C(1:3,:) + [B ; [-4 1]]
```

Exercice 02: (04 points)

Soit la matrice ***M =[0:2;4:-1:2;-4:3:2]***, donner les valeurs de A, B et C après l'exécution de la séquence d'instructions suivantes:

- ***A=M' , B=sum(M), C=diag(M),***

En utilisant la boucle ***for***, proposez une autre séquence d'instructions permettant d'obtenir les même valeurs de A, B et C toujours à partir de M.

Exercice 03: (06 points)

Soit le programme suivant :

```
V=input('Entrez un vecteur:');
G=V(1); P=V(1); x=length(V);
for i=2:x
    if V(i)> P
        P=V(i);
    end
    if V(i)<G
        G=V(i);
    end
end
G  % afficher la valeur de G
P  % afficher la valeur de P
```

1. Exécuter manuellement ce programme pour le vecteur : V = [1, 3, 1, 4, 0]
2. Que fait ce programme ?
3. Remplacer l'instruction ***for*** par ***while*** en préservant la fonctionnalité du programme.
4. Transformer ce programme en une fonction nommée extremum en choisissant les arguments d'entrée et de sortie qu'il faut.

Exercice 04: (05 points)

Soit *A* une matrice carrée de nombres réels. Ecrire le programme (**script**) qui permet de :

1. Lire la matrice *A* (utiliser ***input***);
2. Déterminer le minimum de la dernière ligne ;
3. Construire la matrice B identique à A mais dont la première ligne et la dernière ligne sont permutées.
4. Construire un vecteur *V1* constitué des éléments de l'anti-diagonale de B.
5. Construire un vecteur *V2* constitué des éléments du vecteur *V1* dont le rang (la position) est impair ;
6. Vérifier dans la matrice B s'il y'a des colonnes identiques à des lignes (contiennent les mêmes éléments), afficher alors le numéro de la ligne et de la colonne.

Corrigé Examen de LCS 2014

Solution Exo1 (5 pts)

```
A =  2    0   -2
    -1    2    5
```
0,75

```
B =  1    2
     0    3
```
0,75

```
C = -2
    -2
     3
     5
```
0,75

```
D =  5   -2    1
     0    3   29
```
0,75

```
E =  2    3
     1    4
```
0,75

```
F =  5
    -2
```
0,75

G : Error using +
Matrix dimensions must agree.
0,5

Solution Exo2 (4 pts)

```
M =                    (0.25)
   0    1    2
   4    3    2
  -4   -1    2
```

```
A =
   0    4   -4
(0.25) 1    3   -1
   2    2    2
```

```
B =
   0    3    6
      (0.25)
```

```
C =
(0.25)  0
        3
        2
```

```
clear all
M=[0:2;4:-1:2;-4:3:2]
[n,m]=size(M); (0.5)
B=zeros(1,m);
for j=1:m (01)
  for i=1:n
      A(j,i)=M(i,j); (0.5)

      B(j)=B(j)+M(i,j); (0.5)

  end
  C(j,1)=M(j,j); (0.5)
end
```

Solution1

```
[n,m]=size(M); (0.5)

for i=1:m (0.5)
   for j=1:m
      A(j,i)=M(i,j); (0.5)
   end
end
for j=1:m (0.25)
   B(j)=sum(M(:,j)); (0.5)
end
for j=1:m (0.25)
   C(j,1)=M(j,j); (0.5)
end
```

Solution2

Solution Exo3 (6 pts)

1. V = [1, 3, 1, 4, 0] ⇒ G = 0, P= 4 (01)

2. Le programme **calcule le minimum** du vecteur V et **l'affiche dans G** et **calcule le maximum** de V et **l'affiche dans P.** (01)

3. Remplacement de l'instruction **for** par **while** :

```
V=input('Entrez un vecteur:');
G=V(1); P=V(1); x=length(V);
i=2; (0.5)
while i<=x (0.5)
    if V(i)> P
        P=V(i);
    end
```

1/2

```
    if V(i)<G
        G=V(i);
    end
    i=i+1;
end
G  %afficher la valeur de G
P  % afficher la valeur de P
```

(0.5)

4. Transformation du programme script en fonction nommée ***extremum*** :

```
function [G,P]=extremum(V)
```

(0.25) (0.5) (0.5) (0.25)

```
% V=input('Entrez un vecteur:') → ne pas lire V !
G=V(1); P=V(1); x=length(V);
for i=2:x
    if V(i)> P
        P=V(i);
    end
    if V(i)< G
        G=V(i);
    end
end
% G → ne pas afficher G !
% P → ne pas afficher P !
```

(0.5) (---) (0.5)

Solution Exo4 *(5 pts)*

```
clear, clc
%1
A=input('Introduire les éléments de la matrice carrée A : ')
%2
minDL=min(A(end,:))
%3
B=A;
B(1,:)=A(end,:);
B(end,:)=A(1,:)
%4
V1=diag(fliplr(B))
%5
V2=V1(1:2:end)
%6
n=length(B);
for i=1:n
    for j=1:n
        if B(i,:)==B(:,j)'
            disp('la ligne'),i
            disp('est équivalente à la colonne'),j
        end
    end
end
```

(0,5) (0,5) (0,5) (0,5) (0,5) (0,5) (0,5) (0,5) (0,5)

2 | 2