Université Aberrahmane Mira – Bejaia | Faculté des sciences exactes | Département MI 2017/2018 - S 1

EMD d'une durée de 1h30mn
Module: Outils de programmation 1

Corrigé

Exercice 1: QCM et QCR sur 13 points

Prise en main (sur 3 points)

Q1 : ***Octave*** est

- ❑ **un langage interprété**
- ❑ un langage compilé
- ❑ **spécialisée dans le domaine du calcul matriciel**
- ❑ un clone de ***Scilab***

- ❑ aucune des réponses ci-dessus n'est vraie

Q2 : Un programme écrit en ***C*** est plus rapide qu'un programme écrit en :

- ❑ ***MatLab***
- ❑ ***Octave***
- ❑ ***R***

- ❑ aucune des réponses ci-dessus n'est vraie

Q3 : Donnez la commande octave permettant de supprimer toutes les variables dont le nom commence par « **x** » :

clear x*

Q4 : Que fait la commande « >> **whos ?y*** »

Affiche toutes les variables (fonctions) dont les noms ont « *y* » comme second caractère

Q5 : Que fait la commande « **>> history -c** »

Efface l'historique des commandes

Q6 : Dans le domaine du calcul hautes-performances comme les prévisions météorologiques, il est fortement d'utiliser un langage comme ***Octave*** ou ***Matlab*** du fait des calculs matriciels intenses.

- ❑ ***Faux***

Généralités (sur 3 points)

Q7 : Sur combien de bits sera représentée la variable « **a** » à l'issue de la commande suivante:

« **a = complex(12)** »

Réponse : 64 bits

Q8 : Indiquez la valeur de « **a** » à l'issue des commandes suivante :

« **a = size(12)(1)** »

Réponse : 1

Q9 : Indiquez ce qui sera affiché à l'issue des commandes suivante :

```
Fenêtre de commande
>> alpha =12;
>> clear ?l*
>> beta = alpha +2
```

Réponse : Une erreur car alpha a été supprimée

Q10: Sur combien de bien de bits sera représentée la variable « **a** » à l'issue de la commande suivante :

« **a = double(12) + single(14)** »

Réponse : 32 bits

Q11: Que va afficher Octave, à l'issue des commandes suivantes :

```
Fenêtre de commanc
>> x = sin(0);
>> y = cos(pi/2);
>> sin = x+y;
>> disp(sin(0))
```

Réponse : Une erreur

Q12: Que renvoie la commande suivante :

« **>> 12 - 5i == conj(12 + 5j** »

Réponse :1
(la réponse « erreur » sera acceptée)

Séries (sur 2 points)

Dans les questions suivantes sur les séries, vous devez impérativement utiliser l'opérateur « : » !

Q13 : Donnez la commande octave permettant de créer un vecteur-ligne **V1** composé des valeurs :

π, 2π , 3π , 4 π et 5 π

Réponse : V1 = 1 : pi : 5*pi

Q14 : Donnez la commande octave permettant de créer un vecteur-ligne **V2** composé des valeurs : 0, 5, 10 et15.

Réponse : V2 = 0 : 5 : 15

Q15 : Donnez la commande octave permettant de créer un vecteur-ligne **V3** composé des valeurs : 10, 7, 4 et 1

Réponse : V3 = 10 : -3 : 1

Q16 : Indiquez ce que va contenir « Y » à l'issue des commandes suivantes :

```
Fenêtre de commandes
>> X = (1:5:25);
>> y = X(end:-1:1);
```

Réponse : 21 16 11 6 1

Vecteurs (sur 2 points)

Q17 : Donnez la commande octave permettant de créer un vecteur-ligne **V4** composé des valeurs : 14, 15 et 12:

Réponse : V4 = [14 15 12]

Q18 : Donnez la commande octave permettant de créer un vecteur-colonne **V5** composé des valeurs : 1, 5, et 4:

Réponse : V5 = [1 ; 5 ; 4]

Q19: Donnez la commande octave de créer un vecteur-colonne **V6** composé de 100 valeurs à « 0 »:

Réponse : V6 = zeros(1, 100)

Q20: Que va contenir x, à l'issue des commandes suivantes:

```
Fenêtre de commandes
>> V7 = [1 4 8 10];
>> V7(end) = [];
>> x = size(V7)(2);
```

Réponse : 3

Matrices (sur 3 points)

Q21 – Donnez la commande permettant de créer la matrice **M1** suivante : $M1 = \begin{pmatrix} 1 & 3 & 2 \\ 5 & 2 & 4 \\ 3 & 6 & 6 \end{pmatrix}$

Réponse : M1 = [1 2 3 ; 5 2 4 ; 3 6 6]

Q22 – Je suppose que j'ai crée 2 matrices **A** et **B**. comme suit :

```
Fenêtre de commandes
>> A = [1 2; 3 4; 5 6];
>> B = [1 2 3; 4 5 6];
```

Parmi les opérations ci-dessous, indiquez celles qui sont correctes

- ❑ [~~A B~~]
- ❑ **[A' B]**
- ❑ **[A B']**
- ❑ [~~A' B'~~]

- ❑ Aucune des opérations ci-dessus n'est correcte !

Q23 – Indique ce que va contenir ***P*** à l'issue des commandes suivantes :

```
Fenêtre de commandes
>> M = [1 2 3; 6 7 8];
>> P = M>5;
```

```
>> P = M>5
P =

  0  0  0
  1  1  1
```

Q24 – Donnez la commande permettant de créer une variable « ***p »*** qui va contenir le nombre de colonnes d'une matrice ***M*** :>

Réponse : P = columns(M)

Q25 – Donnez la commande permettant de créer une matrice unitaire « ***M2 »*** à partir de la matrice ***M1*** suivante :

M1

```
 1   2   3   4   5
11  12  13  14  15
21  22  23  24  25
```

Indication : ***M2*** doit être égale à ceci :

```
 2   3
12  13
```

Réponse : M2 = M1(1:2 ; 2:3)

Q26 – Donnez la commande permettant de créer une matrice « **A** » composée de 2 lignes et 3 colonnes de valeurs aléatoires comprises entre 10 et 15 :

Réponse : A = rand(2,3)*5+10

Nom et prénom: ……..…….. **Groupe:** ……..

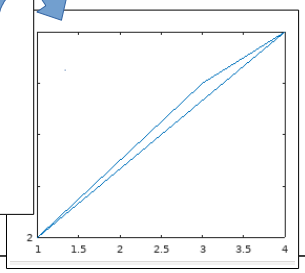
Exercice 2 : Fonctions et scripts sur 7 points

A - Écrire une fonction permettant de dire si une matrice de coordonnées (xi, yi) représente un triangle

Indication :

- Nom de la fonction : « **isTriangle** »
- Paramètre d'entrée : une matrice **M** composée de 2 lignes. Chaque colonne de **M** représente un point de la courbe.
- Usage :

```
Fenêtre de commandes
>> M = [1 3 4 1;
        2 5 6 2];
>> if isTriangle(M)
       X = M(1,:)
       Y = M(2, :)
       plot(X, Y)
   endif
```



2 points

Réponse : ……

```
isTriangle.m
1  function [Rep] = isTriangle(M)
2    Rep = 1
3    nbPoints = columns(M)
4    if nbPoints != 4
5      Rep = 0
6    endif
7    if M(:,1) != M(:,4)
8      Rep = 0
9    endif
10 endfunction
```

0.5 (function [Rep] = isTriangle(M))

0.5 (nbPoints = columns(M))

0.5 (if nbPoints != 4)

0.5 (if M(:,1) != M(:,4))

B - Écrire une fonction « ***racineCarre*** » permettant de retourner un vecteur ***Y*** dont les éléments sont les racines carrés des éléments d'un vecteur ***X*** donné comme paramètre d'entrée

1 point

```
isTriangle.m | racineCarre.m
1  function y = racineCarre(x)
2    y = x.**1/5  ;
3  endfunction
```

0.5 (function y = racineCarre(x))

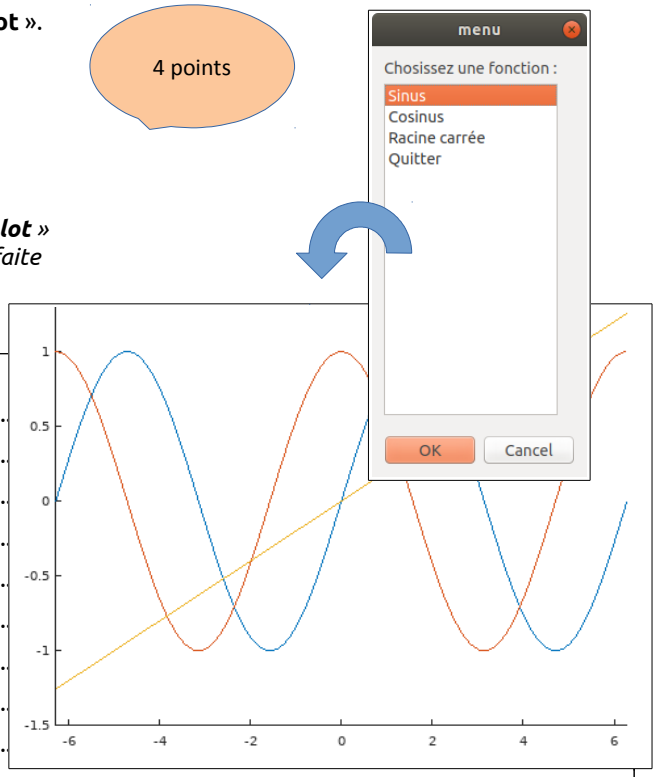
0.5 (y = x.**1/5 ;)

⇨ **C -**Écrire une fonction permettant d'afficher des courbes avec « **fplot** ».

4 points

Indication :

- Nom de la fonction : « **afficherCourbes** »
- Paramètre d'entrée : un intervalle [a, b].
- En sortie :
  - Affiche un menu (voir figure ci-contre)
  - Selon le choix de l'utilisateur affiche une courbe avec « ***fplot*** »
  - *pour la fonction « **racineCarre** », supposez qu'elle est déjà faite (voir exercice précédent)*



affichierCourbe.m

```
1  function affichierCourbe(a,b)
2
3    figure
4    hold on
5
6    while true
7      m = menu( "Chosissez une fonction :",
8                "Sinus",
9                "Cosinus",
10               "Racine carrée",
11               "Quitter")
12     switch m
13       case 1
14         fplot('sin', [a, b])
15       case 2
16         fplot('cos', [a, b])
17       case 3
18         fplot('racineCarre', [a, b])
19       case 4
20         close all
21         break
22     endswitch
23   endwhile
24 endfunction
```

0.5 0.5 0.5 0.5 0.5 0.5 0.5

Bon courage