

EMD 1 Outils de programmation

Corrigé

QCM (sur 6 points)

Q1 : Citez, parmi les fonctions ci-dessous, celles qui peuvent avoir comme argument une matrice:

- real()
- imag()
- conj()
- length()
- single()
- isreal()
- iscomplex()
- isscalar()

Q2 : L'ensemble des commandes, rassemblées dans la figure ci-dessous, constitue une fonction :

- Vrai
- Faux

```

Éditeur
Fichier  Editer  Affichage  Débuguer  Exécuter  Aide
sin2.m ✕
1  % function y = sin2(x)
2  x = input("donnez le vecteur x");
3  y = sin(x).**2;
4  disp(y);
5  % endfunction
6
    
```

Q3 : Que renvoi le code suivant :

```

Fenêtre de commandes
>> sqrt = 10;
>> x = sqrt(4);
    
```

- 0
- 2
- 4
- erreur

Q4 : Pour convertir un nombre (scalaire) vers la représentation de nombre flottant sur 64 bits, on utilise la fonction :

- single
- int64
- double
- float
- real

Q5 : « *na* » veut dire :

- valeur numérique
- not analogique
- not and
- nombre aléatoire
- non disponible

Q6 : L'expression « `2 + int8(eye(3))` » donne :

- un scalaire
- un vecteur-ligne
- un nombre entier
- un vecteur-colonne
- une matrice carrée

Q7 : Je suppose que j'ai créé une variable « *M* » comme suit : « `M = "BAC 2017"` »;. Que va m'indiquer la commande suivante : « `M([end : 1])` »

- BAC 2017
- 2017 BAC
- 2017
- 7102 CAB
- BAC

Aucune
réponse n'est
correcte

La bonne
réponse est une
matrice vide

Q8 : Je suppose que j'ai créé une variable « *N* » comme suit : « `N = [1 4 6 8 9]` »;. Que va m'indiquer la commande suivante : « `N([end : 1, 2])` »

- 1 4 6 8 9
- 1
- 2
- 4
- 6

Q9 : J'ai une matrice *A* ayant **5x3** éléments et une autre matrice *B* ayant **8x3** éléments. Indiquez les constructions correctes :

- `C = [B , A]`
- `C = [B ; A]`
- `C = [A , B]`
- `C = [A ; B]`

Q9 : J'ai une matrice *A* ayant **5x2** éléments et une autre matrice *B* ayant **5x3** éléments. Indiquez les constructions correctes :

- `C = [B , A]`
- `C = [B ; A]`
- `C = [A , B]`
- `C = [A ; B]`

Q11 : L'expression « `size(M)(2)` » retourne.

- Le nombre de colonnes de *M*
- la taille de *M*
- le nombre de ligne de *M*
- la longueur de *M*
- le nombre d'éléments de *M*

Q12 : Le code suivant affiche

```

Fenêtre de commandes
>> A = [eye(2) zeros(2)];
>> disp(sum(sum(A)))
    
```

- 4
- 3
- 2
- 5

Questions à réponses courtes (sur 6 points)

Q1 : A l'issue des commandes Octave ci-dessous, indiquez la valeur qui sera affichée :

```
Fenêtre de commandes
>> A = [1 2 3 ; 4 5 6]
>> A(2,:)=[];
>> disp(sum(A))
```

Réponse : 6

Q2 : A l'issue des commandes Octave ci-dessous, indiquez ce que va contenir la variable C :

```
Fenêtre de commandes
>> A = eye(3);
>> B = diag(A);
>> C = diag(B);
```

Réponse : Matrice unitaire 3x3. (C = A)

Q3 : Donnez la commande octave permettant de créer une matrice A contenant 2 lignes et 3 colonnes de nombres réels tirés au hasard entre 0 et 5 (0 et 5 compris).

Réponse : A = rand(2,3)*5

Q4 : Donnez la commande octave permettant de créer une matrice B contenant 4 lignes et 3 colonnes de nombres entiers tirés au hasard entre 0 et 3 (0 et 3 compris).

Réponse : A = round(rand(4,3)*3)

Q5 : Indiquez le contenu de la variable « c », à l'issue des commandes suivantes :

```
Fenêtre de commandes
>> a = 400;
>> b = int8(a);
>> c = single(b);
```

Réponse : 127

Q6 : A l'issue des commandes suivantes, indiquez ce qui sera affiché :

```
Fenêtre de commandes
>> x=input("donnez x : ");
donnez x : 12
>> disp(class(x))
```

Réponse : double

Q7 : Indiquez ce que va afficher octave, à l'issue des commandes suivantes :

```
Fenêtre de commandes
>> x=input("donnez x : ");
donnez x : 12
>> typeinfo(x)
```

Réponse : scalar

Q8 : Que va contenir la variable « ans » à l'issue des commandes suivantes :

```
Fenêtre de commandes
>> x=input("donnez x : ");
donnez x : [12 14]
>> typeinfo(x)
```

Réponse : Matrix

Q9 : Que va contenir la variable « ans » à l'issue des commandes suivantes :

```
Fenêtre de commandes
>> x=input("donnez x : ", "s");
donnez x : [12 14]
>> typeinfo(x)
```

Indication : choisissez entre : Matrix, double, integer, complex, sq_string, vector.

Réponse : sq_string

Q10 : Indiquez ce que va contenir la variable « x », à l'issue des commandes suivantes :

```
Fenêtre de commandes
>> A = [1:4; 2:5; 3:6]
A =
     1     2     3     4
     2     3     4     5
     3     4     5     6
>> x = sum(A)(1);
```

Réponse : 6

Q11 : Indiquez ce que va contenir la variable « x », à l'issue des commandes suivantes :

```
Fenêtre de commandes
>> A
A =
     1     2     3     4
     2     3     4     5
     3     4     5     6
>> x = sum(diag(A));
```

Réponse : 9

Q11 : Soit la matrice A suivante :

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 5 \\ 3 & 4 & 5 & 6 \\ 10 & 11 & 12 & 13 \end{bmatrix}$$

Donnez la commande octave permettant d'obtenir, à partir de la matrice A, la matrice B suivante qui correspond aux éléments du milieu de A :

$$B = \begin{bmatrix} 3 & 4 \\ 4 & 5 \end{bmatrix}$$

Réponse : B = A(2:3, 2:3)

Fonctions (sur 2 points)

Ecrire une fonction octave nommée « **mat** » qui renvoit, selon la valeur d'un paramètre « **P** », des matrices (ou vecteurs) comme indiqué dans le tableau suivant :

Valeurs de P	Ce que renvoi la fonction
0	Matrice 5x5 éléments initialisés à 0
1	Matrice unité 5x5
2	Matrice 5x5 composée de nombres tirés au hasard entre 2 et 5
3	Un vecteur composé des nombres multiples de 3 compris entre 20 et 60.

Attention, Pour « **P** » ayant une valeur n'appartenant pas à l'ensemble {0, 1, 2, 3} la fonction « **mat** » doit renvoyer la valeur 0.

Réponse :

```

1 function M = mat(P)
2     switch P
3     case 0
4         M = zeros(5,5);
5     case 1
6         M = eye(5,5);
7     case 2
8         M = rand(5,5)*3+2;
9     case 3
10        M = 21:3:60;
11    otherwise
12        M = 0;
13    endswitch
14 endfunction
    
```

Barème :

- Syntaxe : 0.5 point
- Logique 1.5 points

Barème : Chaque réponse sur 0.25 points

Script (sur 3 points)

Ecrivez un petit script Octave permettant :

1. Définir un vecteur **X** composé des nombres compris entre -2π et $+2\pi$. Ces nombres sont équidistants avec un pas de 0,01.
2. Définir un vecteur **Y** tel que chacun de ses éléments y_i est défini par : $y_i = e^{x_i}$
3. Définir un vecteur **Z** tel que chacun de ses éléments z_i est défini par $z_i = \sqrt{e^{-x_i}}$
4. Créer une figure (vide) numérotée à 3
5. Afficher (avec une seule commande) dans la figure numéro 3 les deux courbes

$$y = e^x \quad \text{et} \quad z = \sqrt{e^{-x}}$$

La courbe $y = e^x$ doit être en bleu avec une épaisseur de trait égale à 2.
 La courbe $z = \sqrt{e^{-x}}$ doit être en rouge (l'épaisseur de trait à 1)
6. Créer une figure numérotée à 5
7. Dessiner la courbe 3 D composé des points issues des vecteurs X, Y et Z.
8. Donner à cette courbe le titre « Courbe 2D »
9. Donnez à l'axe des x le label « abscisses »
10. Donnez à l'axe des y le label « ordonnées »
11. Donnez à l'axe des z le label « altitude »
12. Fermer toutes les figures

```

X = -2*pi:0.01:2*pi; % 1. Définir un vecteur X
Y = exp(X); % 2. Définir un vecteur Y
Z = sqrt(exp(-X)); % 3. Définir un vecteur Y
figure(3); % 4. Créer figure 3

% 4. Afficher courbes y=exp(x) et z = sqrt(exp(-x))
plot(X,Y, 'b', 'linewidth', 2);
hold on
plot(X,Z, 'r', 'linewidth', 1);

figure(5); % 6. Créer une figure 5
plot3(X, Y, Z); % 7. Dessiner courbe 3D (X, Y, Z)
title("Coyrbe 3D"); % 8. titre « Courbe 2D »
xlabel("abscisses") % 9. label des x « abscisses »
ylabel("ordonnées") % 10. label des y « ordonnées »
zlabel("altitude") % 11. label des z « altitude »
close all; % 12. Fermer toutes les figures
    
```

Ecrire un script octave qui :

1. Lit, à partir du clavier, les coordonnées d'un point A dans l'espace (3D)
2. Lit, à partir du clavier, les coordonnées d'un point B dans l'espace (3D)
3. Indique :
 - a. si le point A est au dessus ou en dessous du point B (ici on se base sur l'axe Z).
 - b. si A et B sont sur le même plan horizontal (les deux points on la même coordonnée z)

Attention, votre script doit vérifier si les données saisies par l'utilisateur correspondent vraiment à des coordonnées de points dans l'espace (c'est-à-dire 3 composantes réelles pour chaque point).

```

end-exo-fin.m ✕
1  clc
2  % lecture des coordonnées du point A
3  while true
4      A = input("Donnez les coordonnées de A : ");
5      if isvector(A)
6          if columns(A)==3 break
7          endif
8      endif
9      disp("Vous devez données 3 coordonnées ! ")
10 endwhile
11 % lecture des coordonnées du point B
12 while true
13     B = input("Donnez les coordonnées de B : ");
14     if isvector(B)
15         if columns(B)==3 break
16         endif
17     endif
18     disp("Vous devez données 3 coordonnées ! ")
19 endwhile
20 xa = A(1); ya = A(2); za = A(3);
21 xb = B(1); yb = B(2); zb = B(3);
22 % Vérifions la position de A par rapport à B
23 if za<zb disp("A en dessous de B")
24     elseif za>zb disp("A au dessus de B")
25     else
26         disp("A et B sur le meme plan hor

```