

EMD 1 Outils de programmation

Nom et prénom :

Groupe :

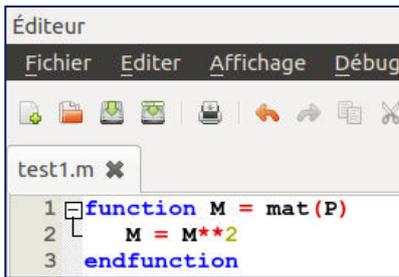
QCM (sur 5 points)

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

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

Q2 : L'ensemble des commandes, rassemblées dans la figure ci-dessous, constituent

- une fonction
- un script
- un code erroné



Q3 : Que renvoi le code suivant :

```
Fenêtre de commandes
>> sin = 1;
>> sin(0);
```

- 0
- 1
- π
- erreur

Q4 : Pour convertir un nombre (scalaire) vers la représentation de nombre flottant sur 32 bits, on utilise la fonction :

- single
- int64
- double
- float
- real

Q5 : « *clear A** » veut dire :

- suppression de tout l'historique
- suppression de toutes les variables
- Suppression de la fenêtre de commande
- Effacement de la fenêtre de commande
- Aucune des réponses ci-dessus n'est bonne

Q6 : L'expression « *4 + int16(zeros(2))* » 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 = "AISA"* »;. Que va m'indiquer la commande suivante : « *M ([end : -1 : 1])* »

- AISA
- ASIA
- SIAA
- Erreur
- Les réponses ci-dessus sont incorrectes

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 :-3:3)* »

- 4
- 1
- 2
- 9
- 8

Q9 : 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 :-3:3)* »

- 4
- 1
- 2
- 9
- 8

Q10 : L'expression « *size(M)(1)* size(M)(2)* » retourne.

- Le nombre de colonnes de M
- le nombre de lignes de M
- la longueur en octet de M
- le nombre d'éléments de M

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
>> M = [1,3 ; 2,0 ;3, 4];
>> disp(numel(M))
```

6

Q2 : A l'issue des commandes Octave ci-dessous, indiquez la valeur qui sera affichée :

```
Fenêtre de commandes
>> M = [1,3 ; 2,0 ;3, 4];
>> disp(rows(M))
```

3

Q3 : A l'issue des commandes Octave ci-dessous, indiquez la valeur qui sera affichée :

```
Fenêtre de commandes
>> M = [1,3 ; 2,0 ;3, 4];
>> disp(size(M)(1))
```

3

Q4 : A l'issue des commandes Octave ci-dessous, indiquez la valeur qui sera affichée :

```
Fenêtre de commandes
>> M = [1,3 ; 2,0 ;3, 4];
>> disp(size(M)(2))
```

2

Q5 : A l'issue des commandes Octave ci-dessous, indiquez ce que va contenir la variable N :

```
Fenêtre de commandes
>> M = [1,3 ; 2,1 ;3, 4];
>> N = diag(diag(M))
```

.....

Q6 : Donnez la commande octave rendre un nombre entier tiré au hasard entre 1 et 10 (1 et 10 compris).

round(rand()*10).....

Q7 : En vous basant sur votre réponse à la question précédente, donnez la commande octave permettant de tirer au hasard un élément d'un vecteur V suivant.

```
Fenêtre de commandes
>> V = [1;4;5;6;0;-3;4;5;7;12]
```

V(round(rand()*10))

Q8 : A l'issue des commandes suivantes, indiquez ce qui sera affiché :

```
Fenêtre de commandes
>> a = input(" ", "s")
[14,15]
a = [14,15]
>> typeinfo(a)
```

- matrix
- double
- integer
- complex
- sq_string
- vector
- scalar

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

```
Fenêtre de commandes
>> a = input("s ");
s 14
>> typeinfo(a)
```

- matrix
- double
- integer
- complex
- sq_string
- vector
- scalar

Soit le système d'équations suivant:

$$\begin{aligned} 4x + 3y + 2z + t &= 1 \\ 3x + 4y + 3z + 2t &= 1 \\ 2x + 3y + 4z + 3t &= -1 \\ x + 2y + 3z + 4t &= -1 \end{aligned}$$

On suppose que l'écriture matricielle de ce système est : $A \cdot X = B$

Q10 : Donnez la commande octave permettant de créer la matrice A du système d'équation précédent

.....

Q11 : Donnez la commande octave permettant de créer le vecteur A du système d'équation précédent

.....

Q12 : Donnez la commande octave permettant de résoudre (trouver le vecteur X) le d'équation précédent

$X = \text{inv}(A) \cdot B$

Ecrire un script octave qui :

1. Lit, à partir du clavier, les coordonnées de 4 points A, B, C et D (**attention, pour faire cette lecture vous allez utiliser la fonction « lireComposantes () » que vous avez défini dans l'exercice précédent**).
2. Vérifie si les 4 points sont situés dans le plan horizontal (ont la même composante z)
3. Vérifie si les 4 points sont situés dans le plan vertical parallèle à l'axe des x (ont la même composante y).

```

lireComposantes.m x * exo3.m x
1  clc
2  % lecture des coordonnées du point A
3  disp("Donnez les composantes de A"); A = lireComposantes();
4
5  % lecture des coordonnées du point B
6  disp("Donnez les composantes de B"); B = lireComposantes();
7
8  % lecture des coordonnées du point C
9  disp("Donnez les composantes de C"); C = lireComposantes();
10
11  xa = A(1); ya = A(2); za = A(3);
12  xb = B(1); yb = B(2); zb = B(3);
13  xc = C(1); yc = C(2); zc = C(3);
14
15  % vérifions si les 3 points sont dans le meme plan horizontal
16  if za==zb && zb==zc
17      disp("les 3 points sont sur le meme plan horizontal")
18  else disp("les 3 points ne sont pas sur le meme plan horizontal")
19  endif
20
21  if ya==yb && yb==yc
22      disp("les 3 points sont sur le plan parallele à x")
23  else disp("les 3 points ne sont pas sur le plan parallele à x")
24  endif

```

Bon courage