

EXAMEN BLANC

Corrigé

QCM (sur 6 points)

**Q1 :** Citez, parmi les fonctions ci-dessous, celles qui n'opèrent pas sur des nombres complexes sous Octave :

- real()
- imag()
- conj()
- abs()
- mod()
- arg()
- isreal()
- iscomplex()

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

- Vrai
- Faux

```

Éditeur
Fichier  Editer  Affichage  Débugger  Exécuter  Ai
gestnotes1.m x
1  clc
2  disp("bienvenue dans la gestion des n")
3
4  n = csvread("csv/notes.data");
5  notes = eliminerNotesAberrantes1(n);
6  disp('notes(1:10,:)');
    
```

**Q3 :** Octave sauvegarde les nombres par défaut sur :

- 8 bits
- 16 bits
- 32 bits
- 64 bits

**Q4 :** Pour convertir un nombre (scalaire) vers la représentation en simple précision, on utilise la fonction :

- simple
- double
- int64
- single
- float
- real

**Q5 :** Lorsque je converti un nombre entier sur 8 bits vers une représentation d'entier sur 64 bits, je gagne en précision :

- Vrai
- Faux

**Q6 :** Le nombre (2 + complex(14)) est un nombre en :

- double précision
- int16
- int32
- int64
- uint32
- complex

**Q7 :** Je suppose que j'ai créé une variable « M » comme suit : « **M = "Analyse proba"** »;. Que va m'indiquer la commande suivante : « M([9 : 11]) »

- Analyse
- proba
- anapro
- Analyse pro
- pro

**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(:, 2) »

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

**Q9 :** J'ai une matrice **N** ayant **4x5** éléments et une autre matrice **P** ayant **5x4** éléments. Je peux construire une nouvelle matrice **Q** en concaténant **N** et **P**

- Verticalement
- Horizontalement
- Je ne peux pas les concaténer

**Q10 :** J'ai une matrice **N** ayant **4x5** éléments et une autre matrice **P** ayant **4x3** éléments. Je peux construire une nouvelle matrice **Q** en concaténant **N** et **P**

- Verticalement
- Horizontalement
- Je ne peux pas les concaténer

**Q11 :** La fonction **size()** retourne.

- un vecteur dont le premier élément est le nombre de lignes
- la taille d'une matrice (une seule valeur)
- un scalaire

**Q12 :** Le code suivant affiche

```

Fenêtre de commandes
>> M = [1:3; 2:4; 3:5; 4:6];
>> disp(size(M)(1))
    
```

- 2
- 3
- 4
- 5
- 6

Questions à réponses courtes (sur 5 points)

Q1 : A l'issue des commandes Octave ci-dessous, indiquez que va représenter la variable Z :

```
Fenêtre de commandes
>> x = ones(5);
>> y = diag(x);
>> z = diag(y);
```

Une matrice unité 5 x 5 .....

.....

.....

Q2 : Que va afficher Octave à l'issue des commandes suivantes :

```
Fenêtre de commandes
>> a = zeros(5);
>> b = ones(3,5);
>> c = a*b;
```

Erreur car le nombre de colonne de a est différent du nombre de ligne de b

.....

Q3 : Donnez la commande octave permettant de créer une matrice **A** contenant 3 lignes et 4 colonnes de nombres réels tirés au hasard entre 0 et 5 (0 et 5 compris).

**A = rand(3,4)\*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 5 (0 et 5 compris).

*Indication : la fonction round permet d'arrondir un nombre réel au nombre entier le plus proche*

**B = round(rand(3,4)\*5)** .....

Q5 : Indiquez ce que va afficher octave, à l'issue des commandes suivantes :

```
Fenêtre de commandes
>> x = 350.5;
>> y = int8(x);
>> z = double(y);
>> disp(z)
```

127 .....

Q6 : A l'issue des commandes suivantes, indiquez la valeur de la variable **b** :

```
Fenêtre de commandes
>> a = 14.55;
>> b = single(int8(a));
```

15 .....

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

```
Fenêtre de commandes
>> a = [12 14];
>> class(a)
```

Double .....

Q8 : Indiquez ce que va afficher octave, à l'issue des commandes suivantes :

```
Fenêtre de commandes
>> a = [12 14];
>> typeinfo(a)
```

Matrix .....

Q9 : Donnez la commande permettant de créer un vecteur-colonne **X** contenant les valeurs impaires allant de 15 à 30:

**X = ([15 :30])'** .....

**X = (15 :30)'** .....

Q10 : Indiquez ce que va afficher octave, à l'issue des commandes suivantes :

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

13 .....

sum(A(1, :))=3  
 sum(A(:, 1))=10  
 [sum(A(1, :)), sum(A(:, 1))] = [3, 10]  
 sum([sum(A(1, :)), sum(A(:, 1))]) = sum([3, 10]) = 13

Q11 : Indiquez ce que va afficher octave, à l'issue des commandes suivantes :

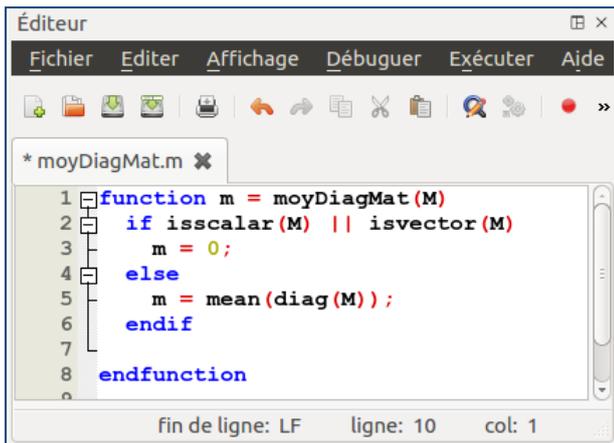
```
Fenêtre de commandes
>> A = [1 1 1
        3 5 7 ] ;
>> mean(A)
```

Me renvoi un vecteur de moyennes chacune correspondant à la moyenne d'une colonne :

**2 3 4** .....

**Fonctions (sur 2 points)**

Ecrire une fonction octave qui calcule la moyenne des éléments de la diagonale d'une matrice  $M$ . Attention, avant de faire ce calcul, votre fonction doit vérifier si  $M$  est un scalaire, un vecteur ou une matrice. Dans le cas où il s'agit d'un scalaire ou d'un vecteur, elle doit renvoyer 0:



```

Éditeur
Fichier Editer Affichage Débuguer Exécuter Aide

*moyDiagMat.m x
1 function m = moyDiagMat(M)
2   if isscalar(M) || isvector(M)
3     m = 0;
4   else
5     m = mean(diag(M));
6   endif
7
8 endfunction
  
```

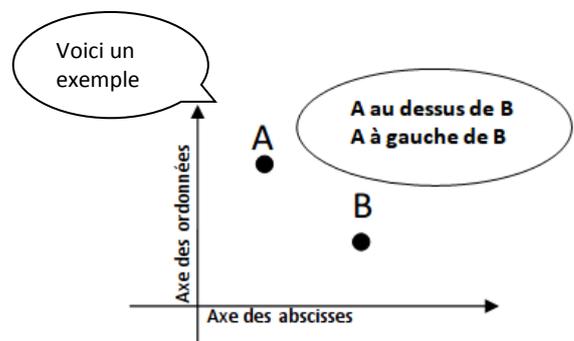
fin de ligne: LF ligne: 10 col: 1

**Script (sur 3 points)**

Ecrire un script octave qui :

1. Lit, à partir du clavier, les coordonnées d'un point A dans le plan
2. Lit, à partir du clavier, les coordonnées d'un point B dans le plan
3. Indique si le point A est au dessus, en dessous, à gauche ou à droite du point B.

Attention, votre script doit vérifier si les données saisies par l'utilisateur correspondent vraiment à des coordonnées de points dans le plan (c'est-à-dire à 2 composantes réelles)



```

xa = ""; ya = ""; xb = ""; yb = "";
while !isscalar(xa)
  xa = input("----> Donnez l'abscisse de A : ");
endwhile

while !isscalar(ya)
  ya = input("----> Donnez l'ordonnée de A : ");
endwhile

while !isscalar(xb)
  xb = input("----> Donnez l'abscisse de B : ");
endwhile

while !isscalar(yb)
  yb = input("----> Donnez l'ordonnée de B : ");
endwhile

if xa < xb
  disp("A est à gauche de B")
endif

if xa > xb
  disp("A est à droite de B")
endif

if ya > yb
  disp("A est au dessus de B")
endif
if ya < yb
  disp("A est en dessous de B")
endif

if (xa == xb) && (ya == yb)
  disp("A est sur B")
endif
  
```

Ecrivez un petit script Octave permettant :

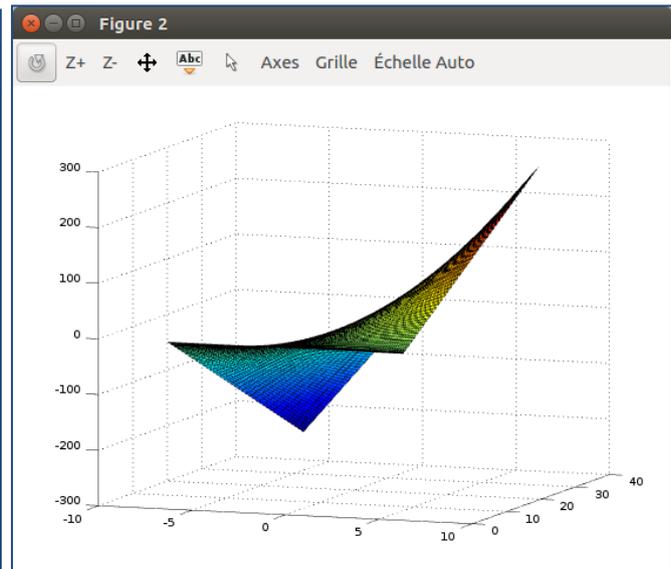
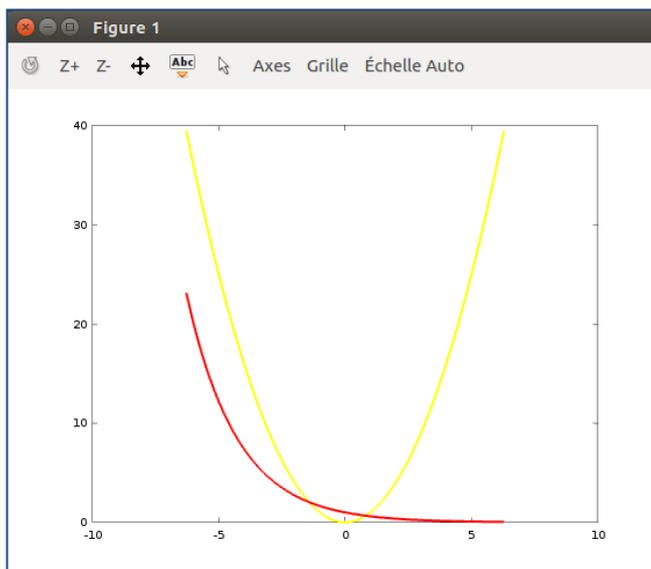
1. Définir un vecteur  $X$  composé de 100 valeurs comprises entre  $-2\pi$  et  $+2\pi$
2. Définir un vecteur  $Y$  tel que chacun de ses éléments sont définis par  $(y_i=x_i^2)$
3. Définir un vecteur  $Z$  tel que chacun de ses éléments  $z_i$  sont définis par  $z_i = (\sqrt{e^{-x_i}})$
4. Afficher sur la même figure les deux courbes  $y = x^2$  et  $z = \sqrt{e^{-x}}$   
La courbe  $y = x^2$  doit être en jaune  
La courbe  $z = \sqrt{e^{-x}}$  doit être en rouge
5. Construit une grille de points  $[X1, Y1]$  à partir des vecteurs  $X$  et  $Y$ . (indication : utilisez `meshgrid`)
6. Affiche une surface à base des vecteurs  $X1, Y1$  et  $Z1$  tel que les éléments de  $Z1$  sont le produit élément par élément des vecteurs  $=X1$  et  $Y1$ .

```

1  clc
2  %1- Définir un vecteur X composé de 100 valeurs comprises entre -2pi et +pi
3  X = linspace(-2*pi, 2*pi, 100);
4
5  %2- Définir un vecteur Y tel que chacun de ses éléments sont définis par (yi=xi^2)
6  Y = X.**2;
7
8  %3- Définir un vecteur Z tel que chacun de ses éléments zi sont définis par zi= (sqrt(e^(-xi)))
9  Z = sqrt(exp(-X));
10
11 %4- Afficher sur la même figure les deux courbes y=x^2 et z=sqrt(e^(-x))
12 %La courbe y=x^2 doit être en jaune
13 %La courbe z=sqrt(e^(-x)) doit être en rouge
14 figure;
15 plot(X,Y, 'y', 'linewidth', 2, X, Z, 'r','linewidth', 2);
16
17 %5- Construit une grille de points [X1, Y1] à partir des vecteurs X et Y.
18 % (indication : utilisez meshgrid)
19
20 [X1, Y1] = meshgrid(X,Y);
21
22 %6- Affiche une surface à base des vecteurs X1, Y1 et Z1 tel que les éléments de Z1
23 % sont le produit élément par élément des vecteurs =X1 et Y1.
24 Z1 = X1.*Y1;
25 figure(2);
26 surf(X1, Y1, Z1);

```

Voici le résultat de l'exécution de ce script :



*Bon courage*