

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

```

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 nombre 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 crée 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 crée 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

```

>> 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-dessus, indiquez que va représenter la variable Z :

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

.....  
.....  
.....

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

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

.....

**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*

.....

**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)
```

.....

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

```
Fenêtre de commandes
>> a = input("donnez a : ")
donnez a : 14.5
a = 14.500
>> b = single(int8(a));
```

.....

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

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

.....

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

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

.....

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

.....

**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)
```

.....



