

Nom et prénom : Groupe :

Introduction (1 point)

Q1 : Des langages interprétés génèrent des programmes plus rapides que ceux générés par des compilateurs :

- Vrai
- Faux

Q2 : Si je veux écrire un programme pour faire du calcul statistique, je privilégie R à Matlab :

- Oui
- Non

Q3 : Python est un langage à usage général, mais dispose de module spécialisés en mathématique, ce qui fait de lui un sérieux concurrent à Matlab :

- Oui
- Non

Q4 : Python est un langage à usage général, mais dispose de module spécialisés en mathématique, ce qui fait de lui un sérieux concurrent à Octave :

- Oui
- Non

Prise en main (1 point)

Q5 : Citez 4 fenêtres de l'interface graphique d'Octave (Attention, votre réponse doit être complète pour qu'elle soit considérée comme juste !):

- Fenêtre de commande
- Explorateur de fichier
- Espace de travail+ historique des commandes
- Editeur + documentation

Q6 : Donnez la commande permettant d'avoir de l'aide sur la fonction « log() »:

- help log

Q7 : Donnez la commande permettant d'afficher l'historique des commandes:

- history

Q8 : Donnez la commande permettant d'effacer l'historique des commandes:

- history -c

Généralités (2 points)

Q9 : Indiquez la valeur de la variable « ans » à l'issue de l'exécution de la commande suivante :

```
>> typeinfo(sin(12))
```

- scalar

Q10 : Indiquez la valeur de la variable « ans » à l'issue de l'exécution de la commande suivante :

```
>> typeinfo([1 2 ; 4 3])
```

- matrix

Q11 : Que renvoi la fonction « namelengthmax »:

- nombre maxi de caractère pour un identificateur

Q12 : Lorsque je saisis l'instruction suivante : « a = 12 ». La variable « a » sera représentée sur 8 bits :

- Vrai
- Faux

Q13 : A l'issue des commandes ci-dessous, indiquez ce qui sera affiché :

```
>> x = 13;
>> clear all
>> disp(x);
```

- Erreur

Q14 : Dans la représentation « simple précision », Octave utilise :

- 16 bits
- 32 bits
- 64 bits

Scalaire, séries, vecteurs et matrices (3.5 points)

Q15 – La fonction `isscalar()` permet d'indiquer

- Si une variable est un vecteur
- Si une variable est un nombre
- Si une variable est une matrice
- Si une variable est un scalaire

Q16 – La commande « `x = 10 :-1 :2` » permet de :

- Créer un vecteur-ligne composé d'une ligne et de zéro colonne (donc vecteur vide)
- Créer un vecteur-ligne composé des scalaires allant de 10 à 2
- Créer un vecteur-ligne composé des scalaires allant de 2 à 10

Q17 – La commande « `x = 1 : 0.5 : 3` » permet de créer un vecteur-ligne composé des scalaires suivants:

- 1.0000 1.5000 2.0000 2.5000 3.0000
- 1.0000 2.0000 3.0000
- 2.0000 3.0000 1.0000

Q18 – Pour créer un vecteur-ligne on utilise comme séparateur des valeurs:

- L'espace
- La tabulation
- La virgule
- Le point-virgule
- La touche « ENTER »

Q19 – Donnez la commande permettant de créer le vecteur-colonne composé des valeurs suivantes : 1, 5, 6 et 10.

`[1; 5; 6; 10]`

Q20 – A l'issue des commandes suivantes :

```
Fenêtre de commandes
>> V1 = 1:2:8;
>> V2 = V1';
>> V3 = [1 4 6 7];
>> V4 = [2 4 5 6]';
>> V5 = [4 ; 6];
```

Indiquez si les vecteurs V1 à V5 sont des vecteurs-ligne ou colonne ?

V1 : Vecteur ligne

V2 : Vecteur colonne

V3 : Vecteur ligne

V4 : Vecteur colonne

V5 : Vecteur colonne

Q21 – A l'issue des commandes ci-contre, que va contenir V ?

`0 2 0 4 0`

```
Fenêtre de commandes
>> V = 1:5;
>> for i=1:2:length(V)
    V(i)=0
end;
```

Q22 – Donnez la commande permettant de supprimer les valeurs situées dans les positions impaires d'un vecteur V

`V(1:2:end) = []`

Ou : `V(end:-2:1) = []`

Q23 – La concaténation verticale de vecteurs-colonnes de dimensions différentes est possible

- Vrai
- Faux

Q24 – Que doit vérifier 2 matrices pour pouvoir les concaténer verticalement

Avoir le même nombre de colonnes

Q25 – Que doit vérifier une matrice à n lignes et p colonnes pour pouvoir lui concaténer un vecteur-ligne verticalement

Nombre de colonnes du vecteur = p

Q26 – Donnez la commande Octave me permettant de créer une matrice unité 5x5

`eye(5)` ou `eye(5,5)`

Q27 – Je suppose que j'ai exécuté les commandes Octave suivantes :

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

Parmi les commandes octave ci-dessous, indiquez celles qui sont valides:

- `C = A' * B`
- `C = A * B`
- `C = B * A`
- `C = B' * A`

Q28 – Je suppose qu'on a créé la matrice A suivante :

$$A = \begin{bmatrix} 16 & 2 & 3 & 13 \\ 5 & 11 & 10 & 8 \\ 9 & 7 & 6 & 12 \\ 4 & 14 & 15 & 1 \end{bmatrix}$$

Parmi les commandes ci-dessous, indiquez celles qui permettent d'obtenir la matrice B suivante :

$$B = \begin{bmatrix} 16 & 2 \\ 5 & 11 \\ 9 & 7 \\ 4 & 14 \end{bmatrix}$$

- `B = A(:, 0:2)`
- `B = A(0:4, 0:2)`
- `B = A(:, 1:2)`
- `B = A(1:4, 1:2)`

Chapitre 4 : programmation (sur 1.5 points)

Q29 – Que va afficher le code suivant :

```
Fenêtre de commandes
>> v = input("donnez une valeur : ");
donnez une valeur : 14
>> if isscalar(v)
    disp("1")
else
    disp("2")
end
```

0.25

1

Q30 – Que va afficher le code suivant :

```
Fenêtre de commandes
>> v = input("donnez une valeur : ");
donnez une valeur : [12+5i 14+12i; 15 14]
>> if ismatrix(v) & iscomplex(v)
    disp("1")
elseif ismatrix(v)
    disp("2")
else
    disp(3)
end
```

0.25

1

Q 31 (sur 1 point) – Ecrivez une fonction permettant de retourner la transposée d'une matrice A et cela sans utiliser l'opérateur Octave permettant de donner la transposée d'une matrice.

```
function At = transposer(A)
    for i = 1:rows(A)
        for j = 1:columns(A)
            At(j,i)=A(i,j)
        endfor
    endfor
endfunction
```

0.25

Chapitre 5 : Générer des graphiques (sur 1.5 points)

Q 32 – Ecrivez un petit script Octave permettant :

- Définir un vecteur X composé de 100 valeurs comprises entre -2π et $+2\pi$
- Définir un vecteur Y tel que chacun de ses éléments sont définis par $(y_i=x_i^2)$
- Définir un vecteur Z tel que chacun de ses éléments z_i sont définis par $z_i = \sqrt{e^{-x_i}}$
- 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

```
X = linspace(-2*pi, 2*pi, 100);
Y = X.**2;
Z = sqrt(exp(-X));
plot(X, Y, 'y');
hold on;
plot(X, Z, 'r');
```

0.25

0.25

0.25

0.25

0.25

0.25