

Exercice 1. Tests sur la création de vecteurs et de matrices

On suppose que n un entier supérieur ou égal à 2 a été créé numériquement (prenez par exemple $n = 5$).

1. Écrivez une commande permettant de créer une liste de -3 à n avec un pas de 2.

```
--> n=5;u=-3:2:n
u =

    -3.    -1.     1.     3.     5.
```

2. Écrivez une commande permettant de créer une liste de pas constant de -3 à n et comportant 6 points.

```
--> n=5;u=linspace(-3,n,6)
u =

    -3.    -1.4    0.2    1.8    3.4    5.
```

3. Quelle est la différence entre les commandes suivantes ?

```
u=n:3:1 et linspace(n,1,3)

--> n=5;u=n:3:1,v=linspace(n,1,3),
u =

    5.    3.    1.

[] // on va de 5 à 1 avec un pas croissant de 3, ce qui n'est pas possible...
v =

    5.    3.    1.
```

4. Quel est le vecteur renvoyé par la commande suivante ?

```
u=n:-n

--> n=5;u=n:-n,
u =

    5.    3.    1.    -1.    -3.

[] // on va de 5 à -5 avec un pas croissant de 1, ce qui n'est pas possible...
```

5. Quelle est la matrice renvoyée par la suite d'instructions qui suit ?

```
u=1:3; v=ones(1,3); w=-1:2:4; x=[u;v;w]

--> u=1:3; v=ones(1,3); w=-1:2:4; x=[u;v;w]
x =

    1.    2.    3.
    1.    1.    1.
   -1.    1.    3.
```

Exercice 2. Création efficace de vecteurs à l'aide d'opérations arithmétiques pointées

1. Écrire une seule commande permettant de créer le vecteur $x = \left(5, \frac{5}{2}, \frac{5}{3}, \frac{5}{4}, \dots, \frac{5}{10}\right)$ sans saisir un à un les éléments.

```
--> x=5*ones(1,10)./[1:10]
x =

    5.    2.5    1.6666667    1.25    1.    0.8333333    0.7142857    0.625    0.5555556
    0.5
```

2. Même question pour le vecteur $y = \left(1, \frac{1}{4}, \frac{1}{9}, \frac{1}{16}, \frac{1}{25}, \dots, \frac{1}{100}\right)$.

```
--> y=ones(1,10)./([1:10].^2)
y =

    1.    0.25    0.1111111    0.0625    0.04    0.0277778    0.0204082    0.015625
    0.0123457    0.01
```

3. Même question pour le vecteur $z = (1, 2, 4, \dots, 2^{10})$.

```
--> z=2.^[0:10]
```

```
z =
```

```
1.      2.      4.      8.     16.     32.     64.     128.     256.     512.     1024.
```

Exercice 3. Création de matrices

Écrire une ligne de commandes permettant de créer la matrice de $\mathcal{M}_n(\mathbb{R})$ dont les éléments diagonaux sont égaux à a et les autres éléments égaux à b , pour a , b et n entrés par l'utilisateur.

Comme on a

$$\begin{pmatrix} a & b & \dots & \dots & b \\ b & a & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & a & b \\ b & \dots & \dots & b & a \end{pmatrix} = b \begin{pmatrix} 1 & 1 & \dots & \dots & 1 \\ 1 & 1 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & 1 & 1 \\ 1 & \dots & \dots & 1 & 1 \end{pmatrix} + (a-b) \begin{pmatrix} 1 & 0 & \dots & \dots & 0 \\ 0 & 1 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & 1 & 0 \\ 0 & \dots & \dots & 0 & 1 \end{pmatrix}$$

```
--> n=input('donnez n'); a=input('donnez a'); b=input('donnez b'); A=b*ones(n,n)+(a-b)*eye(n,n)
```

```
donnez n 3
```

```
donnez a 2
```

```
donnez b 1
```

```
A =
```

```
2.      1.      1.
1.      2.      1.
1.      1.      2.
```

Exercice 4. Min et Max

Scilab sait trouver le plus grand élément (c'est la fonction `max`) et le plus petit élément (c'est la fonction `min`) d'un vecteur ou d'une matrice.

1. Écrire une commande permettant de déterminer le plus grand multiple de 7 inférieur ou égal à 1000.

```
--> max(0:7:1000)
```

```
ans =
```

```
994.
```

2. Écrire une commande permettant de déterminer le plus petit multiple de 7 supérieur ou égal à 1000.

```
--> max(0:7:1000)+7
```

```
ans =
```

```
1001.
```

3. Écrire une commande permettant d'entrer trois entiers naturels n , p et q puis de déterminer le plus petit entre le plus grand multiple de p inférieur ou égal à n et le plus grand multiple de q inférieur ou égal à n .

```
-->n=input('donnez n'); p=input('donnez p'); q=input('donnez q'); x=max(0:p:n),y=max(0:q:n),
```

```
z=min(x,y)
```

```
donnez n 5000
```

```
donnez p 11
```

```
donnez q 13
```

```
x =
```

```
4994.
```

```
y =
```

```
4992.
```

```
z =
```

```
4992.
```

Exercice 5. Autour de sum et cumsum

On suppose que n a été créé numériquement.

1. Écrire une ligne de commandes qui renvoie la somme des n premiers entiers naturels non nuls.

```
--> n=10; x=1:n; s=sum(x)
s =

55.
```

2. Écrire une ligne de commandes renvoyant la somme $\sum_{k=1}^n \frac{1}{k}$.

```
--> n=10; x=1:n; y=ones(1,n); t=sum(y./x)
t =

2.9289683
```

3. Que calculent les commandes suivantes ?

(a) `x=ones(1,n); y=cumsum(x)`

```
--> n=10; x=ones(1,n); y=cumsum(x)
y =
```

1. 2. 3. 4. 5. 6. 7. 8. 9. 10.

Pour $n \in \mathbb{N}^*$, y est la liste des n premiers entiers non nuls.

(b) `x=ones(1,n); y=sum(cumsum(x))`

```
--> n=10; x=ones(1,n); y=sum(cumsum(x))
y =
```

55.

Pour $n \in \mathbb{N}^*$,

$$y = 1 + 2 + \cdots + n = \sum_{i=1}^n i = \frac{n(n+1)}{2} \quad (\text{somme d'une suite arithmétique})$$

(c) `x=ones(1,n); y=sum(cumsum(cumsum(x)))`

```
--> n=10; x=ones(1,n); z=cumsum(cumsum(x)), y=sum(z)
z =
```

1. 3. 6. 10. 15. 21. 28. 36. 45. 55.

y =

220.

Pour $n \in \mathbb{N}^*$, z est la liste

$$(1, 1+2, 1+2+3, \dots, 1+2+\cdots+n)$$

On a donc $y = 1 + (1+2) + (1+2+3) + \cdots + (1+2+\cdots+n)$, soit

$$y = \sum_{k=1}^n \sum_{i=1}^k i = \sum_{k=1}^n \frac{k(k+1)}{2} = \frac{1}{2} \left(\sum_{k=1}^n k^2 + \sum_{k=1}^n k \right) = \frac{1}{2} \left(\frac{n(n+1)(2n+1)}{6} + \frac{n(n+1)}{2} \right) = \frac{n(n+1)(n+2)}{6}$$

Exercice 6. La boucle for

On vient de voir que l'on pouvait écrire $\sum_{k=1}^n \frac{1}{k}$ avec la commande `sum`. Sans utiliser la commande `sum`, proposer une suite de commandes utilisant une boucle `for` et permettant de calculer, pour un entier naturel donné la valeur de

$$u_n = \left(\sum_{k=1}^n \frac{1}{k} \right) - \ln(n).$$

Tester pour $n = 100$, $n = 1000$, $n = 10000$.

```
n=input('donnez n');
s=0;
for k=1:n
    s=s+1/k;
end;
u=s-log(n) //
```

Pour $n = 100$, $u_n = 0.5822073$.

Pour $n = 1000$, $u_n = 0.5777156$.

Pour $n = 10000$, $u_n = 0.5772657$.

La limite de cette suite est appelée constante d'Euler et vaut $\gamma \approx 0.577216$.

Exercice 7. La boucle while

Écrire une ligne de commandes permettant de déterminer le plus petit entier naturel n pour lequel

$$\sum_{k=1}^n \frac{1}{k} > 10$$

```
k=0; s=0;
while s<=10
    k=k+1;
    s=s+1/k;
end
disp(k,'k=')
```

k=

12367.

Exercice 8. Suite récurrente

On définit la suite (u_n) par $u_0 \in \mathbb{R}$ et pour $n \in \mathbb{N}$

$$u_{n+1} = \frac{1}{2} \sqrt{3 + u_n^2}.$$

1. Écrire une ligne de commandes demandant une valeur de n et u_0 et permettant de calculer le terme u_n .

```
u=input('donnez u0');
n=input('donnez n');
for k=1:n
    u=sqrt(3+u^2)/2;
end;
disp(u)
```

2. Ajouter une commande permettant de calculer la somme $\sum_{k=0}^n u_k$.

```
u=input('donnez u0');
n=input('donnez n');
s=u;
for k=1:n
    u=sqrt(3+u^2)/2;
    s=s+u;
end;
disp(s)
```

3. On admet que la suite (u_n) converge et que sa limite vaut 1. Écrire une suite de commandes qui permette de déterminer le plus petit entier naturel n pour lequel $|u_n - 1| \leq 10^{-3}$, u_0 étant entré par l'utilisateur.

```
u=input('donnez u0');
n=0;
while abs(u-1) > 10^(-3)
    n=n+1;
    u=sqrt(3+u^2)/2;
end;
disp(n,'n=')
```