

Nom :

Prénom :

Exercice 1. QCM.

Dans ce questionnaire à choix multiples, une seule réponse est correcte. **Cochez la bonne réponse.**
Barème indicatif : réponse correcte 1 pt; pas de réponse 0 pt; réponse fausse -0.5 pt

(Q1) Parmi ces instructions, laquelle permet de tirer un entier aléatoirement entre -8 et 0 (inclus)?

- `random(-8, 0)`
- `randint(-8, 0)`
- `randint(-8, 1)`
- `random(-8, 1)`

(Q2) Dans python, on suppose qu'on a importé la fonction `sin` de la bibliothèque `math`. Que retourne l'instruction suivante?

```
sin(0)
```

- le nombre flottant 0.0
- l'entier 0

(Q3) Dans le logiciel Sagemath, que retourne l'expression suivante?

```
2 // 8
```

- l'entier 0
- le rationnel $1/4$
- le nombre 0.25
- l'expression $2/8$

(Q4) Qu'affiche la série d'instructions suivante?

```
L = [ -3, 7, 1 ]  
print( max([ j-5 for j in L ]) )
```

- 7
- 2
- 1
- l'expression $7 - 5$

Exercice 2. Questions courtes.

Question 1.– Dans Sagemath, on suppose qu'un entier n a déjà été déclaré. Donner l'instruction qui calcule $8! + \binom{n}{4}$.

```
1 factorial(8) + binomial(n, 4)
```

Question 2.– Donner l'instruction qui stocke le nombre complexe $2i + 4$ dans une variable `a`.

```
1 a = 2*I + 4
```

Question 3.– Dans python, donner l'instruction qui permet d'importer la fonction sinus de la bibliothèque `math`.

```
1 from math import sin
```

Question 4.– Dans Sagemath, soit `L` une liste de nombres complexes. Donner l'instruction qui, **en une ligne**, stocke dans une variable `M` la liste des parties imaginaires des éléments de `L`.

```
1 M = [ z.imag_part() for z in L ]
```

Exercice 3. Valuation paire.

Question 1.– Écrire une fonction `valuation_paire(n)` qui prend en entrée un entier naturel n , et qui retourne le plus grand entier naturel k tel que n est divisible par 2^k . Votre fonction **devra** utiliser une boucle `while`.

Par exemple, sur l'entrée $n = 24$, la fonction retournera 3, car $2^3 = 8$ divise 24, mais $2^4 = 16$ ne divise pas 24.

```
1 def valuation_paire(n):
2     val = 0
3     while n % (2**val) == 0:
4         val += 1
5     return val-1
```

Exercice 4. Plus grande partie réelle.

Question 1.– On utilise le logiciel Sagemath. Écrire une fonction `plus_grande_partie_reelle(L)` qui prend en entrée une liste non-vide de nombres complexes L , et qui retourne l'élément de L ayant la plus grande partie réelle. Si plusieurs éléments de la liste ont la même partie réelle, on retournera le premier de ces éléments dans l'ordre de la liste.

Par exemple, sur l'entrée $L = [1+I, -2-3*I, 4+2*I, 7*I]$, la fonction retournera $4+2*I$.

```
1 def plus_grande_partie_reelle(L):
2     resultat = L[0]
3     reelle = resultat.real_part()
4     for element in L:
5         if element.real_part() > reelle:
6             resultat = element
7             reelle = resultat.real_part()
8     return resultat
```

Nom :

Prénom :

Exercice 1. QCM.

Dans ce questionnaire à choix multiples, une seule réponse est correcte. **Cochez la bonne réponse.**
Barème indicatif : réponse correcte 1 pt ; pas de réponse 0 pt ; réponse fausse -0.5 pt

(Q1) Parmi ces instructions, laquelle permet de tirer un entier aléatoirement entre 7 et 11 (inclus) ?

- `random(7, 11)`
- `randint(7, 11)`
- `randint(7, 12)`
- `random(7, 12)`

(Q2) Dans python, on suppose qu'on a importé la fonction `cos` et le nombre `pi` de la bibliothèque `math`. Que retourne l'instruction suivante ?

```
cos(pi)
```

- le nombre flottant -1.0
- l'entier -1

(Q3) Dans le logiciel Sagemath, que retourne l'expression suivante ?

```
-2 / 8
```

- l'entier 0
- le rationnel $-1/4$
- le nombre -0.25
- l'expression $-2/8$

(Q4) Qu'affiche la série d'instructions suivante ?

```
L = [ -3, 7, 1 ]  
print( min([ i*2 for i in L ]) )
```

- -3
- -6
- 0
- l'expression $-3 * 2$

Exercice 2. Questions courtes.

Question 1.– Dans Sagemath, on suppose qu'un entier n a déjà été déclaré. Donner l'instruction qui calcule $\binom{n}{5} - 3!$.

```
1 binomial(n, 5) - factorial(3)
```

Question 2.– Donner l'instruction qui stocke le nombre complexe $-4 + 7i$ dans une variable u .

```
1 u = - 4 + 7 * I
```

Question 3.– Dans python, donner l'instruction qui permet d'importer la fonction tangente de la bibliothèque `math`.

```
1 from math import tan
```

Question 4.– Dans Sagemath, soit L une liste de nombres rationnels. Donner l'instruction qui, **en une ligne**, stocke dans une variable M la liste des dénominateurs des éléments de L .

```
1 M = [ r.denominator() for r in L ]
```

Exercice 3. Racine cubique entière.

Question 1.– Écrire une fonction `racine_cubique_entiere(n)` qui prend en entrée un entier naturel n , et qui retourne le plus grand entier naturel k tel que $k^3 \leq n$. Votre fonction **devra** utiliser une boucle `while`.

Par exemple, sur l'entrée $n = 66$, la fonction retournera 4, car $4^3 = 64 \leq 66 < 5^3 = 125$.

```
1 def racine_cubique_entiere(n):
2     k = 0
3     while k**3 <= n:
4         k += 1
5     return k-1
```

Exercice 4. Plus petite partie imaginaire.

Question 1.– On utilise le logiciel Sagemath. Écrire une fonction `plus_petite_partie_imaginaire(L)` qui prend en entrée une liste non-vide de nombres complexes L , et qui retourne l'élément de L ayant la plus petite partie imaginaire. Si plusieurs éléments de la liste ont la même partie imaginaire, on retournera le premier de ces éléments dans l'ordre de la liste.

Par exemple, sur l'entrée $L = [1+I, -2-3*I, 4+2*I, 7*I]$, la fonction retournera $-2-3*I$.

```
1 def plus_petite_partie_imaginaire(L):
2     resultat = L[0]
3     imag = resultat.imag_part()
4     for element in L:
5         if element.imag_part() < imag:
6             resultat = element
7             imag = resultat.imag_part()
8     return resultat
```

Nom :

Prénom :

Exercice 1. QCM.

Dans ce questionnaire à choix multiples, une seule réponse est correcte. **Cochez la bonne réponse.**

Barème indicatif : réponse correcte 1 pt; pas de réponse 0 pt; réponse fausse -0.5 pt

(Q1) Parmi ces instructions, laquelle permet de tirer un entier aléatoirement entre 0 et 19 (inclus)?

- `random(0, 19)`
- `randint(0, 19)`
- `randint(0, 20)`
- `random(0, 20)`

(Q2) Dans python, on suppose qu'on a importé la fonction `sqrt` de la bibliothèque `math`. Que retourne l'instruction suivante?

```
sqrt(4)
```

- le nombre flottant 2.0
- l'entier 2

(Q3) Dans le logiciel Sagemath, que retourne l'expression suivante?

```
6 / 4
```

- l'entier 1
- le rationnel 3/2
- le nombre 1.5
- l'expression 6/4

(Q4) Qu'affiche la série d'instructions suivante?

```
L = [ -3, 7, 1 ]  
print( max([ 2*i for i in L ]) )
```

- 7
- 14
- 1
- l'expression $2 * 7$

Exercice 2. Questions courtes.

Question 1.– Dans Sagemath, on suppose qu'un entier n a déjà été déclaré. Donner l'instruction qui calcule $7! + \binom{n}{3}$.

```
1 factorial(7) + binomial(n, 3)
```

Question 2.– Donner l'instruction qui stocke le nombre complexe $3 - 4i$ dans une variable v .

```
1 v = 3 - 4 * I
```

Question 3.– Dans python, donner l'instruction qui permet d'importer la fonction exponentielle de la bibliothèque `math`.

```
1 from math import exp
```

Question 4.– Dans Sagemath, soit L une liste de nombres rationnels. Donner l'instruction qui, **en une ligne**, stocke dans une variable M la liste des numérateurs des éléments de L .

```
1 M = [ r.numerator() for r in L ]
```

Exercice 3. Taille binaire.

Question 1.– Écrire une fonction `taille_binaire(n)` qui prend en entrée un entier naturel n , et qui retourne le plus petit entier naturel k tel que $2^k > n$. Votre fonction **devra** utiliser une boucle `while`.

Par exemple, sur l'entrée $n = 17$, la fonction retournera 5, car $2^4 = 16 \leq 17 < 2^5 = 32$.

```
1 def taille_binaire(n):
2     k = 0
3     while 2**k <= n:
4         k += 1
5     return k
```

Exercice 4. Plus grand dénominateur.

Question 1.– On utilise le logiciel Sagemath. Écrire une fonction `plus_grand_dénominateur(L)` qui prend en entrée une liste non-vide de nombres rationnels L , et qui retourne l'élément de L ayant le plus grand dénominateur. Si plusieurs éléments de la liste ont le même dénominateur, on retournera le premier de ces éléments dans l'ordre de la liste.

Par exemple, sur l'entrée $L = [1/3, -2/15, 3/4, -1/2]$, la fonction retournera $-2/15$.

```
1 def plus_grand_dénominateur(L):
2     resultat = L[0]
3     den = resultat.denominator()
4     for element in L:
5         if element.denominator() > den:
6             resultat = element
7             den = resultat.denominator()
8     return resultat
```

Nom :

Prénom :

Exercice 1. QCM.

Dans ce questionnaire à choix multiples, une seule réponse est correcte. **Cochez la bonne réponse.**
Barème indicatif : réponse correcte 1 pt; pas de réponse 0 pt; réponse fausse -0.5 pt

(Q1) Parmi ces instructions, laquelle permet de tirer un entier aléatoirement entre -4 et 12 (inclus)?

- `random(-4, 12)`
- `randint(-4, 12)`
- `randint(-4, 13)`
- `random(-4, 13)`

(Q2) Dans python, on suppose qu'on a importé la fonction `exp` de la bibliothèque `math`. Que retourne l'instruction suivante?

```
exp(0)
```

- le nombre flottant 1.0
- l'entier 1

(Q3) Dans le logiciel Sagemath, que retourne l'expression suivante?

```
6 // 4
```

- l'entier 1
- le rationnel $3/2$
- le nombre 1.5
- l'expression $6/4$

(Q4) Qu'affiche la série d'instructions suivante?

```
L = [ -3, 7, 1 ]  
print( min([ j-5 for j in L ]) )
```

- -3
- -8
- 0
- l'expression $-3 - 5$

Exercice 2. Questions courtes.

Question 1.– Dans Sagemath, on suppose qu'un entier n a déjà été déclaré. Donner l'instruction qui calcule $10! - \binom{n}{6}$.

```
1 factorial(10) - binomial(n, 6)
```

Question 2.– Donner l'instruction qui stocke le nombre complexe $1 - 3i$ dans une variable z .

```
1 z = 1 - 3 * I
```

Question 3.– Dans python, donner l'instruction qui permet d'importer la fonction cosinus de la bibliothèque `math`.

```
1 from math import cos
```

Question 4.– Dans Sagemath, soit L une liste de nombres complexes. Donner l'instruction qui, **en une ligne**, stocke dans une variable M la liste des parties réelles des éléments de L .

```
1 M = [ z.real_part() for z in L ]
```

Exercice 3. Factorielle inverse.

Question 1.– Écrire une fonction `factorielle_inverse(n)` qui prend en entrée un entier naturel n , et qui retourne le plus petit entier naturel k tel que $k! > n$. Votre fonction **devra** utiliser une boucle `while`, et **pourra** utiliser la fonction de factorielle de Sagemath.

Par exemple, sur l'entrée $n = 27$, la fonction retournera 5, car $4! = 24 \leq 27 < 5! = 120$.

```
1 def factorielle_inverse(n):
2     k = 0
3     while factorial(k) <= n:
4         k += 1
5     return k
```

Exercice 4. Plus petit numérateur.

Question 1.– On utilise le logiciel Sagemath. Écrire une fonction `plus_petit_numerateur(L)` qui prend en entrée une liste non-vide de nombres rationnels L , et qui retourne l'élément de L ayant le plus petit numérateur. Si plusieurs éléments de la liste ont le même numérateur, on retournera le premier de ces éléments dans l'ordre de la liste.

Par exemple, sur l'entrée $L = [-2/15, 1/3, 3/4, -5/2]$, la fonction retournera $-5/2$.

```
1 def plus_petit_numerateur(L):
2     resultat = L[0]
3     num = resultat.numerator()
4     for element in L:
5         if element.numerator() < num:
6             resultat = element
7             num = resultat.numerator()
8     return resultat
```