

Test d'entraînement (énoncés)

Question 1. Parmi les objets Python définis ci-dessous, lesquels sont mutables ?

- ☐ `[1,1,0]`
- ☐ `(3,4)`
- ☐ `{1,2}`
- ☐ `frozenset([0,1])`
- ☐ `"bonjour"`

Question 2. On considère la fonction Python suivante, qui prend en entrée une liste de nombres et renvoie un nombre :

```
def max_prod(L):  
    n = len(L)  
    m = 0  
    for i in range(n):  
        for j in range(i):  
            if L[i]*L[j]>m:  
                m = L[i]*L[j]  
    return m
```

Si n désigne le nombre d'éléments de la liste L , quelle est la complexité de cet algorithme ?

- ☐ $O(1)$
- ☐ $O(\log n)$
- ☐ $O(n)$
- ☐ $O(n \log n)$
- ☐ $O(n^2)$
- ☐ $O(2^n)$
- ☐ $O(e^n)$

Question 3. On considère une fonction Python f définie avec l'en-tête ci-dessous :

```
def f(x,y=1,z="hello"):  
    ...
```

Quels sont, parmi les appels ci-dessous, ceux qui produiront nécessairement une erreur ?

- ☐ `f(3)`
- ☐ `f("hello",2)`
- ☐ `f(y=8,x=2)`
- ☐ `f(3,5,y=2)`
- ☐ `f(0,y="hello",z=1)`
- ☐ `f(y=1)`

Question 4. Qu'affiche le programme ci-dessous ?

```
I = lambda f,x:f(x+1)-f(x)  
I(lambda x:x**2,5)
```

Question 5. Si A est une matrice carrée, combien de multiplications matricielles sont-elles nécessaires pour calculer A^{29} par exponentiation rapide ?

Question 6. Déterminer le type de l'objet Python produit par l'instruction considérée :

- a) `[(x**2,) for x in range(10)]`
- b) `(x**2 for x in range(10))`
- c) `(sum([x**2 for x in range(10)]))`
- d) `[sum(x**2 for x in range(10))]`
- e) `(sum([x**2]) for x in range(10))`
- f) `(sum(x**2 for x in range(10)),)`

choix possibles: nombre, tuple, liste, générateur, ensemble, dictionnaire

Question 7. Que vaut la file F après exécution de l'algorithme suivant ?

```
P <- pile_vide()
F <- file_vide()
pour n=1,2,3,4
    empiler n dans P
tant que P n'est pas vide
    dépiler x de P
    ajouter x à F
    extraire y de F
    ajouter y à F
```

Donner les éléments de la file F du dernier au premier, sous la forme de nombres séparés par des virgules, sans espace.

Question 8. Donner un pseudo-code pour une fonction f qui prend en argument un entier n et retourne la liste de tous les entiers entre 1 et n (inclus) qui sont des carrés parfaits. Par exemple, $f(20)$ doit retourner la liste $[1,4,9,16]$

Question 9. Donner un pseudo-code pour une fonction récursive f qui prend en argument la racine d'un arbre dont les étiquettes sont des nombres, et retourne la valeur maximale parmi les étiquettes de tous les nœuds de l'arbre. Ce pseudo-code pourra utiliser les fonctions de base `étiquette()` et `enfants()`, dont les signatures sont :

```
étiquette() : noeud -> nombre
enfants() : noeud -> ensemble de nœuds
```

Question 10. Donner un pseudo-code pour une fonction f qui prend en entrée un ensemble de nombres A et retourne l'ensemble de tous les couples (x, y) avec $x \in A$, $y \in A$ et $x < y$.

Question 11. Traduire le pseudo-code ci-dessous en une fonction Python (on supposera qu'une fonction `pgcd()` a été préalablement écrite en Python):

```
fonction f(n)
    // retourne un graphe simple non orienté dont les sommets sont les entiers de 1 à n
    // et deux entiers a et b sont voisins si pgcd(a,b)>1
    // le graphe est codé par un dictionnaire
    D <- dictionnaire vide
    pour i allant de 1 à n
        E <- ensemble vide
        pour j allant de 1 à n
            si j est différent de i et pgcd(i,j)>1
                ajouter j à E
            ajouter à D la clé i, avec comme valeur correspondante l'ensemble E
    retourner D
```

Question 12. Traduire le pseudo-code ci-dessous en une fonction génératrice Python :

```
fonction génératrice f(s)
    // retourne un générateur de tous les sous-ensembles de N*
    // dont la somme des éléments vaut s
    vérifier que s est de type entier et que s>0
    délivrer l'ensemble {s}
    pour k=1,2,...,s-1
        pour tout E délivré par f(s-k)
            si la valeur maximale de E est strictement inférieure à k
                ajouter k à E
                délivrer E
```

Question 13. Traduire le pseudo-code ci-dessous en une fonction récursive Python :

```
fonction inversions(L)
    // retourne le nombre d'indices i,j tels que i<j et L(i)>L(j)
    si L est vide
        retourner 0
    x <- L(0)
    s <- inversions( L privé de son premier élément L(0) )
    s <- s + nombre d'éléments y de L tels que y<x
    retourner s
```

Question 14. Écrire un code Python qui calcule et affiche le plus petit nombre entier $n \geq 0$ tel que

$$n^3 - 19n^2 \geq 1351.$$

Question 15. Écrire une fonction Python `occurrences()` qui prend en entrée une liste de nombres `L` et renvoie un dictionnaire `D` tel que pour toute valeur `x` présente dans `L`, `D[x]` contient l'indice de la dernière occurrence de `x` dans `L`. Par exemple, l'appel `occurrences([1,3,2,1,5,2,1,5])` renvoie le dictionnaire

$$\{1:6, 2:5, 3:1, 5:7\}$$

Question 16. Écrire en Python une fonction `limite()` qui prend en entrée un argument `x` de type `float` et renvoie la limite quand $n \rightarrow +\infty$ de x^n (et `None` si la limite n'existe pas).

Question 17. Écrire en Python une fonction génératrice `f()` qui prend en entrée une chaîne de caractères `s` et délivre (dans un ordre quelconque) toutes les sous-chaînes formées de 2 caractères consécutifs de `s`.

Par exemple, l'appel `f("Bonjour")` doit renvoyer successivement les valeurs `'Bo'`, `'on'`, `'nj'`, `'jo'`, `'ou'`, `'ur'`

Question 18. Écrire une fonction **récursive** `f(a,n)` qui prend en entrée deux entiers naturels a et n , et renvoie le terme u_n de la suite définie par $u_0 = a$ et la récurrence

$$\forall n \in \mathbb{N}, \quad u_{n+1} = \begin{cases} u_n - 1 & \text{si } u_n \text{ est impair,} \\ \frac{u_n}{2} & \text{si } u_n \text{ est pair.} \end{cases}$$

Par exemple, `f(13,1)` doit renvoyer 12, `f(13,2)` doit renvoyer 6 et `f(13,5)` doit renvoyer 1

Question 19. Écrire une fonction non réursive $\mathbf{f}(\mathbf{n})$ qui prend en entrée un entier naturel $n \geq 3$ et renvoie la liste $[u_1, u_2, \dots, u_n]$, où $(u_n)_{n \geq 1}$ est la suite définie par

$$\forall n \in \mathbb{N}^*, \quad u_n = \begin{cases} 1 & \text{si } n \leq 3, \\ u_{n-2} + u_{n-3} & \text{si } n \geq 4. \end{cases}$$

Par exemple, $\mathbf{f}(10)$ doit renvoyer la liste $[1, 1, 1, 2, 2, 3, 4, 5, 7, 9]$
