

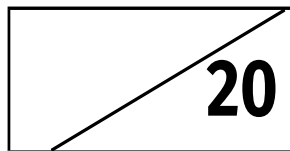
Interrogation d'Informatique

du 25 au 29/5/2026

Nom :		Prénom :	
-------	--	----------	--

Consignes

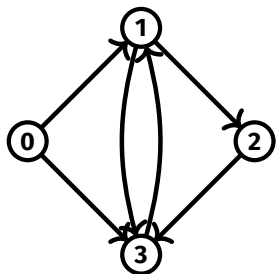
- En Mathématiques, les énoncés du cours doivent être complètement écrits : un cadre, des hypothèses et une conclusion.
- En Informatique : les scripts doivent être correctement indentés, en mettant en valeur l'indentation à l'aide d'une barre verticale.
- La note finale tiendra compte, directement ou indirectement, de la qualité de la rédaction et de la présentation.
- Le crayon à papier ne sera pas corrigé. ● L'usage de la calculatrice est interdit.



GRAPHES

Exercice 1 | Dictionnaire et graphes [\[Solution\]](#)

1. 2 On considère le graphe ci-dessous. Écrire le dictionnaire associé en Python.



2. 2 On considère le dictionnaire ci-dessous, tracer le graphe associé.

$G = \{0 : [1], 1 : [0, 2, 3], 2 : [1, 3, 4], 3 : [1, 2, 4], 4 : [2, 3]\}$



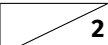
- 1 Ce graphe est-il orienté?



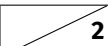
3. 2 Écrire une fonction `nb_arcs(G)` qui prend en argument un dictionnaire codant un graphe et renvoyant le nombre d'arcs (c'est-à-dire le nombre d'arêtes orientées) présentes dans le graphe.



STATISTIQUES

1.  Soit une série statistique codée par une liste L que l'on suppose triée dans l'ordre croissant. Écrire une fonction d'en-tête `mediane(L)` qui renvoie la médiane de la série.




2.  Soit une série statistique codée par une liste L . Écrire une fonction d'en-tête `modalites(L)` qui renvoie la liste des modalités de la série.



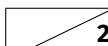
ALÉATOIRE

Exercice 2 | [\[Solution\]](#)

1.  Écrire une fonction, en langage Python, nommée `aLea` qui prend en argument un entier positif k et renvoie 1 avec la probabilité $\frac{k+1}{k+2}$ et 0 avec la probabilité $\frac{1}{k+2}$. Cela revient donc à simuler une bernoulli de paramètre $\frac{k+1}{k+2}$...



2. Un mobile se déplace sur les points à coordonnées entières d'un axe selon les règles suivantes :
- À l'instant $n = 0$, le mobile est au point d'abscisse 0.
 - Si à l'instant $n \in \mathbb{N}$, le mobile est au point d'abscisse k , alors à l'instant $n + 1$, il est au point d'abscisse $k + 1$ avec probabilité $\frac{k+1}{k+2}$ et reste au point d'abscisse k avec probabilité $\frac{1}{k+2}$.

On note X_n l'abscisse du mobile à l'instant n .  Écrire une fonction, en langage Python, nommée `simuX` qui prend en entrée un entier naturel n non nul et qui simule n déplacements du mobile et renvoie la valeur de X_n .



Exercice 3 | [Solution] Les deux questions de cet exercice sont indépendantes

1.

2

 Écrire une fonction Python `simuPF` prenant en argument un entier `N` et qui simule une série de `N` lancers d'une pièce équilibrée, et qui renvoie **la liste** des résultats de ces lancers ("Pile" est codé par 1, et "Face" par 0).



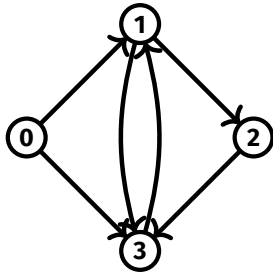
2.

3

 Écrire une fonction Python `simu` sans argument qui simule une série de lancers d'une pièce équilibrée jusqu'à l'obtention de la configuration "Pile, Pile, Face", et renvoie **le nombre** de lancers nécessaires à l'apparition de cette configuration.

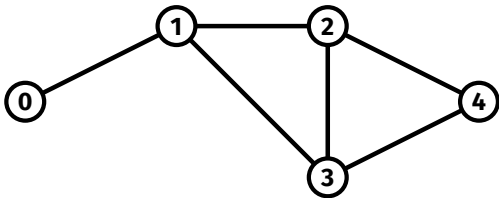
Solution (exercice 1) [Énoncé]

- On considère le graphe ci-dessous. Écrire le dictionnaire associé en Python.



Son dictionnaire associé est :

- On considère le dictionnaire ci-dessous, tracer le graphe associé.
 $G = \{0 : [1], 1 : [0, 2, 3], 2 : [1, 3, 4], 3 : [1, 2, 4], \backslash$
 $\leftarrow 4 : [2, 3]\}$
 Sa représentation associée est :



Le graphe n'est pas orienté.

- Voir TP.

Solution (exercice 2) [Énoncé]

- ```
def alea(k):
 p = (k+1)/(k+2)
 if rd.random() < p:
 return 1
 else:
 return 0
```

```
>>> alea(2)
```

```
1
```

- ```
def simulX(n):
```

```
X = 0
for _ in range(n):
    X += alea(X)
return X
```

```
>>> simulX(10)
```

```
8
```

Solution (exercice 3) [Énoncé]

- ```
def simuPF(N):
 L = []
 for _ in range(N):
 if rd.random() < 1/2:
 L.append(1)
 else:
 L.append(0)
 return L
```
- ```
def simu():
    L = simuPF(3)
    while L[-3:] != [1, 1, 0]:
        if rd.random() < 1/2:
            L.append(1)
        else:
            L.append(0)
    return len(L)
```

```
>>> simuPF(10)
```

```
[0, 1, 1, 0, 0, 0, 1, 1, 1, 1]
```

```
>>> simu()
```

```
8
```