

# BACCALAURÉAT

SESSION 2024

---

Épreuve de l'enseignement de spécialité

## NUMÉRIQUE et SCIENCES INFORMATIQUES

Partie pratique

Classe Terminale de la voie générale

---

Sujet n°26

---

DURÉE DE L'ÉPREUVE : 1 heure

**Le sujet comporte 3 pages numérotées de 1 / 3 à 3 / 3  
Dès que le sujet vous est remis, assurez-vous qu'il est complet.**

*Le candidat doit traiter les 2 exercices.*

## EXERCICE 1 (10 points)

Écrire une fonction `ajoute_dictionnaires` qui prend en paramètres deux dictionnaires `d1` et `d2` dont les clés et les valeurs associées sont des nombres et renvoie le dictionnaire `d` défini de la façon suivante :

- les clés de `d` sont celles de `d1` et celles de `d2` réunies ;
- si une clé est présente dans les deux dictionnaires `d1` et `d2`, sa valeur associée dans le dictionnaire `d` est la somme de ses valeurs dans les dictionnaires `d1` et `d2` ;
- si une clé n'est présente que dans un des deux dictionnaires, sa valeur associée dans le dictionnaire `d` est la même que sa valeur dans le dictionnaire où elle est présente.

Exemples :

```
>>> ajoute_dictionnaires({1: 5, 2: 7}, {2: 9, 3: 11})
{1: 5, 2: 16, 3: 11}
>>> ajoute_dictionnaires({}, {2: 9, 3: 11})
{2: 9, 3: 11}
>>> ajoute_dictionnaires({1: 5, 2: 7}, {})
{1: 5, 2: 7}
```

## EXERCICE 2 (10 points)

On considère une piste carrée qui contient 4 cases par côté. Les cases sont numérotées de 0 inclus à 12 exclu comme ci-dessous :

0	1	2	3
11			4
10			5
9	8	7	6

L'objectif de l'exercice est d'implémenter le jeu suivant :

Au départ, le joueur place son pion sur la case 0. A chaque coup, il lance un dé équilibré à six faces et avance son pion d'autant de cases que le nombre indiqué par le dé (entre 1 et 6 inclus) dans le sens des aiguilles d'une montre.

Par exemple, s'il obtient 2 au premier lancer, il pose son pion sur la case 2 puis s'il obtient 6 au deuxième lancer, il le pose sur la case 8, puis s'il obtient à nouveau 6, il pose le pion sur la case 2.

Le jeu se termine lorsque le joueur a posé son pion sur **toutes les cases** de la piste.

Compléter la fonction `nombre_coups` ci-dessous de sorte qu'elle renvoie le nombre de lancers aléatoires nécessaires pour terminer le jeu.

Proposer ensuite quelques tests pour en vérifier le fonctionnement.

```
from random import randint
```

```
def nombre_coups():
    '''Simule un jeu de plateau avec 12 cases et renvoie le nombre
    minimal de coups pour visiter toutes les cases.'''
    nombre_cases = 12
    # indique si une case a été vue
    cases_vues = [ False ] * nombre_cases
    nombre_cases_vues = 1
    cases_vues[0] = True
    case_en_cours = 0
    n = ...
    while ... < ...:
        x = randint(1, 6)
        case_en_cours = (case_en_cours + ...) % ...
        if ...:
            cases_vues[case_en_cours] = True
            nombre_cases_vues = ...
    n = ...
    return n
```