

# **BACCALAURÉAT**

**SESSION 2026**

---

**Épreuve de l'enseignement de spécialité**

## **NUMÉRIQUE et SCIENCES INFORMATIQUES**

**Partie pratique**

**Classe Terminale de la voie générale**

---

**Sujet n°14**

---

**DURÉE DE L'ÉPREUVE : 1 heure**

**Le sujet comporte 4 pages numérotées de 1 / 4 à 4 / 4  
Dès que le sujet vous est remis, assurez-vous qu'il est complet.**

Cette situation d'évaluation comporte ce document ainsi que des fichiers de codes et de données présents sur l'ordinateur à la disposition du candidat. Le candidat doit restituer ce document avant de sortir de la salle d'examen. Le candidat doit agir en autonomie et faire preuve d'initiative tout au long de l'épreuve.

En cas de difficulté, le candidat peut solliciter l'examineur afin de lui permettre de continuer la tâche. Des moments privilégiés pour solliciter l'examineur sont indiqués dans le document sous la forme d'appels professeur.

L'examineur peut intervenir à tout moment, s'il le juge utile.

Lors de la construction d'un bâtiment, d'un lieu culturel ou sportif, le respect des normes de sécurité amène à se poser la question du nombre judicieux de sorties, de leurs emplacements et du temps nécessaire pour l'évacuation totale des occupants.

Ce sujet propose de finaliser une application permettant de simuler l'évacuation d'une pièce rectangulaire. Cette pièce sera une instance de la classe `Piece` dont le code est dans le fichier `simulation_evacuation.py` du dossier fourni. Le constructeur de cette classe permet de définir la profondeur et la largeur de la pièce.

La méthode `ajouter_occupants(i, j, nb)` permet d'ajouter jusqu'à `nb` occupants dans la case située ligne `i` et colonne `j`, sachant que le nombre d'occupants d'une case est obligatoirement compris entre 0 et 5.

La méthode `ajouter_sortie(direction, position)` permet d'ajouter une sortie à la pièce bien que, pour l'instant, seules les directions "N" (pour le nord) et "O" (pour l'ouest) soient prises en compte. Lors de l'affichage d'une pièce, les sorties sont représentées par la lettre P.

Voici un exemple d'utilisation de cette classe. Le programme

```
p1 = Piece(5, 7)
p1.ajouter_occupants(2, 0, 4)
p1.ajouter_occupants(3, 4, 1)
p1.ajouter_occupants(0, 5, 2)
p1.ajouter_sortie("N", 5)
print(p1)
```

produit l'affichage console

```

      P
[0, 0, 0, 0, 2, 0]
[0, 0, 0, 0, 0, 0]
[4, 0, 0, 0, 0, 0]
[0, 0, 0, 1, 0, 0]
[0, 0, 0, 0, 0, 0]
```

La méthode `alerter` permet de simuler une alerte : chaque occupant essaie de se rapprocher d'une sortie en se déplaçant d'une case (vers le nord, le sud, l'est ou l'ouest) ; chaque sortie ne laisse passer qu'une seule personne par alerte.

Voici, par exemple, trois alertes successives sur la pièce précédente. Il n'est pas nécessaire de comprendre, ni de modifier, le code de cette méthode.

	P		P		P
[0, 0, 0, 0, 0, 1, 0]	[0, 0, 0, 0, 0, 0, 0]	[0, 0, 0, 0, 0, 0, 0]			
[0, 0, 0, 0, 0, 0, 0]	[0, 4, 0, 0, 0, 0, 0]	[0, 0, 4, 0, 0, 1, 0]			
[0, 4, 0, 0, 0, 1, 0]	[0, 0, 0, 0, 0, 1, 0]	[0, 0, 0, 0, 0, 0, 0]			
[0, 0, 0, 0, 0, 0, 0]	[0, 0, 0, 0, 0, 0, 0]	[0, 0, 0, 0, 0, 0, 0]			
[0, 0, 0, 0, 0, 0, 0]	[0, 0, 0, 0, 0, 0, 0]	[0, 0, 0, 0, 0, 0, 0]			

### Question 1

Écrire le corps de la méthode `nb_occupants_restants` de la classe `Piece`. Comme son nom l'indique, cette méthode doit renvoyer le nombre d'occupants restants dans la pièce. La fonction `test_nb_occupants_restants` présente dans le fichier `simulation_evacuation.py` vous permettra d'effectuer une première série de tests.



Appeler le professeur pour lui présenter votre réponse ou en cas de difficulté.

### Question 2

Écrire le corps de la fonction `evacuation` afin qu'elle simule l'évacuation complète de la pièce et renvoie le nombre de tours nécessaire. On pourra, dans cette fonction, faire appel à la méthode `alerter` qui simule un tour et renvoie `True` si des déplacements ont pu avoir lieu, `False` sinon. En complément de la pièce à évacuer, la fonction `evacuation` a un paramètre silencieux dont la valeur par défaut est `True`. Si ce paramètre vaut `False`, l'état de la pièce doit être affiché à chaque tour dans la console. La fonction `test_evacuation` présente dans le fichier `simulation_evacuation.py` vous permettra d'effectuer une série de tests.



Appeler le professeur pour lui présenter votre réponse ou en cas de difficulté.

### Question 3

Modifier la méthode `ajouter_sortie(direction, position)` afin qu'il soit aussi possible d'ajouter une sortie dans les directions qui ne sont pour l'instant pas prises en compte : "S" (pour le sud) et "E" (pour l'est). La fonction `test_ajouter_sortie` vous permettra d'effectuer une première série de tests. Vous vérifierez également qu'il est maintenant possible d'ajouter des sorties dans les quatre directions via l'interface homme machine (IHM), sans modifier le code de celle-ci. Lorsqu'une pièce a été créée dans l'IHM, un clic en périphérie de cette pièce déclenche automatiquement un appel à la méthode `ajouter_sortie` et fait apparaître la porte ajoutée. Cependant, une seule porte sera utilisée lors des alertes tant que la question suivante n'aura pas été traitée.



Appeler le professeur pour lui présenter votre réponse ou en cas de difficulté.

On s'aperçoit que, lorsqu'une pièce possède plusieurs sorties, seule la première est utilisée par les occupants. Le problème vient de la méthode `choix_sortie(i, j)` qui renvoie la sortie à

utiliser pour une personne positionnée sur la ligne  $i$  et la colonne  $j$ .

#### Question 4

Identifier l'erreur logique et la variable non définie dans le code de cette méthode, puis effectuer les corrections nécessaires afin qu'elle renvoie la sortie la plus proche.

La fonction `test_choix_sortie` vous permettra d'effectuer une première série de tests. Vous poursuivrez vos tests avec l'IHM (sans la modifier).



Appeler le professeur pour lui présenter votre réponse ou en cas de difficulté.

### Description du dossier

Le dossier fourni au candidat sur l'ordinateur comporte les éléments suivants : - une version PDF de l'énoncé ; - un code source à compléter et corriger `simulation_evacuation.py` ; - un programme `IHM_evacuation.py` permettant d'ouvrir une IHM qui facilitera les tests de l'élève, à utiliser sans modification.

### Préparation de l'environnement

Pour faire fonctionner le code fourni dans le dossier, les bibliothèques suivantes doivent être présentes : `random`, `copy`, `tkinter`.