

2026 sujet 14

Question 1

```
def nb_occupants_restants(self):  
    ''' renvoie le nombre d'occupants restants dans la pièce.  
        A FAIRE (QUESTION 1)  
    '''  
    nb = 0  
    for i in range(self.i_max + 1):  
        for j in range(self.j_max + 1):  
            nb = nb + self.grille[i][j]  
    return nb
```

Question 2

```
def evacuation(p, silencieux=True):  
    ''' simule l'évacuation de la pièce et renvoie le nombre de tours nécessaire.  
        A chaque tour, chacun des occupants se déplace, si possible, d'une case  
        vers la sortie la plus proche. Si le paramètre silencieux vaut false,  
        l'état de la pièce à chaque tour est affiché dans la console.  
        A FAIRE EN QUESTION 2  
    '''  
    tour = 0  
    while p.alerter(silencieux):  
        tour += 1  
    return tour
```

Question 3

Une sortie est déterminée par les coordonnées de la case où elle se situe.

```
def ajouter_sortie(self, direction, position):  
    ''' permet d'ajouter des sorties à la pièce.  
        A COMPLETER (QUESTION 3) (Pour l'instant, on n'utilise que deux directions !)  
    '''  
    if direction == "N":  
        self.sorties.append((0, position))  
    elif direction == "O":  
        self.sorties.append((position, 0))  
    elif direction == "S":  
        self.sorties.append((self.i_max, position))  
    elif direction == "E":  
        self.sorties.append((position, self.j_max))
```

L'IHM semble bien fonctionner.

Question 4

Il manque le calcul de la nouvelle distance et la comparaison dans le `if` n'est pas bonne.

```
def choix_sortie(self, i, j):  
    ''' renvoie la sortie à utiliser pour une personne positionnée sur la ligne i e  
        A CORRIGER (QUESTION 4) (Pour l'instant, seule la 1ère sortie est utilisée  
    '''  
    assert len(self.sorties) > 0, "Aucune sortie"  
    choix = self.sorties[0]  
    distance = abs(i - choix[0]) + abs(j - choix[1])  
    for k in range(1, len(self.sorties)):  
        autre_sortie = self.sorties[k]  
        d2 = abs(i - autre_sortie[0]) + abs(j - autre_sortie[1])  
        if d2 < distance:  
            choix = autre_sortie  
            distance = d2  
    return choix
```

L'évacuation fonctionne alors correctement dans l'IHM.