

PSI**DS d'informatique N°1****Nom :****Prénom :****Document Réponses****Note :****Exercice 1 : Base de données d'hôtel**

Q1 - Lister les numéros des chambres de l'hôtel ayant au moins trois couchages avec leur nombre de couchages.

Q2 - Donner la capacité moyenne des chambres de l'hôtel.

Q3 - Lister les étages et leur nombre de chambres.

Q4 - Quel sont les étages qui ont une capacité moyenne des chambres supérieures ou égales à 2,5 ?
Lister les étages et leur capacité moyenne des chambres.

Q5 - Lister les noms et prénoms des clients ayant donné une adresse personnelle à PARIS.

Q6 - Lister les noms et prénoms des clients qui ont donné une adresse personnelle et une adresse professionnelle.

Q7 - Donner la liste des personnes (Nom Prénom et numéro de téléphone) qui occupaient une chambre au premier étage de l'hôtel le 11 septembre 2025 ('2025-09-11')

Exercice 2 : Optimisation de la furtivité de la trajectoire d'un drône

Q1 - A quoi correspond le dictionnaire retourné par cette fonction ?

Q2 - Compléter le code de la fonction `def dic_risques_type2(zone:list) -> dict` (lignes 4 à 6) qui retourne le dictionnaire de tous les déplacements élémentaires de type 2.

```
def dic_risques_type2(zone):
    Dico={}
    p,q=len(zone)-1,len(zone[0])-1

    return Dico
```

Q3 - Compléter le code de la fonction `def dic_risques_type3(zone:list) -> dict` (lignes 4 à 6) qui retourne le dictionnaire de tous les déplacements élémentaires de type 3.

```
def dic_risques_type3(zone):
    Dico={}
    p,q=len(zone)-1,len(zone[0])-1

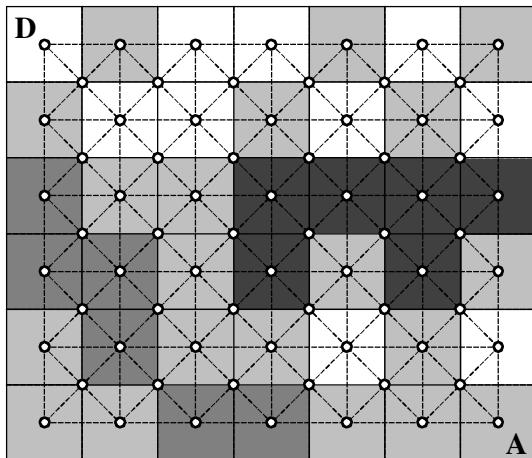
    return Dico
```

Q4 - Compléter le code de la fonction `def dic_risques_type5(zone:list) -> dict` (lignes 4 à 6) qui retourne le dictionnaire de tous les déplacements élémentaires de type 5.

```
def dic_risques_type5(zone):
    Dico={}
    p,q=len(zone)-1,len(zone[0])-1

    return Dico
```

- Q5 -** Tracer sur le document réponse la trajectoire liée à la liste `['E', 'E', 'E', 'E', 'E', 'SE', 'SE', 'S', 'S', 'S']` puis calculer le risque lié à cette trajectoire.



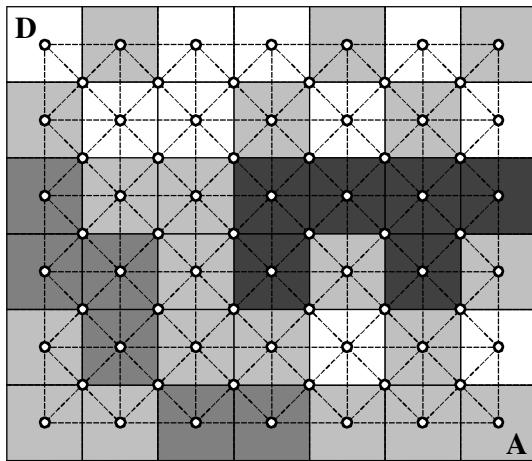
Risque liée à la trajectoire :

- Q6 -** Implémenter la fonction `calcul_risque_trajectoire(L_directions :list, DR_deplac :dict) -> float` qui prend en argument la liste `L_directions` d'une trajectoire, le dictionnaire `DR_deplac` des risques des déplacements élémentaires et qui retourne le risque lié à cette trajectoire.

- Q7 -** Quelle serait la complexité d'un algorithme utilisant cette stratégie ? Pourquoi va-t-on exclure cette stratégie algorithmique ?

Q8 - De quel type d'algorithme relève cette stratégie ?

Q9 - Tracer sur le document réponse la trajectoire du drône obtenue avec cette stratégie. Donner également le risque total résultant de votre trajectoire (utiliser $\sqrt{2} \approx 1,4$). Remarque : Il peut y avoir plusieurs réponses possibles, une seule est demandée.



Risque liée à la trajectoire :

Q10 - Implémenter une fonction `direction_moins_risquee(DR_deplac:dict, i:int,j:int,p:int,q:int) -> str` qui prend en argument un dictionnaire `DR_deplac` des risques de tous les déplacements élémentaires une position donnée par les entiers `i` et `j`, la dimension de la zone $((2.p+1) \times (2.q+1))$ donnée par les entiers `p` et `q`, et qui retourne la direction ('`E`', '`S`' ou '`SE`') correspondant au déplacement élémentaire dont le risque est minimal.

Q11 - Compléter le code de cette fonction `strategie1(zone:list)`.

```
def strategie1(zone):
    DR_deplac=dic_risques(zone)
    trajectoire=[]
    i,j=0,0
    p,q=len(zone)-1,len(zone[0])-1
```

Q12 - Cette stratégie est-elle optimale ? Justifier la réponse.

Q13 - De quel type d'algorithme relève cette stratégie ?

Q14 - Que retourne cette fonction ? Donner le type des éléments retournés.

Q15 - Quelles sont les lignes du code qui reprennent l'équation de Bellman de l'algorithme.

Q16 - De quel type est la sous fonction `traj_opt(i, j)` ?

Q17 - Qu'est-ce qui justifie cela. Justifier votre réponse à partir du code de cette fonction donné ci-dessus.

Q18 - Implémenter une fonction **strategie2_memoisation(zone)** similaire à la fonction **strategie2(zone)** mais qui a une complexité linéaire. Vous utiliserez pour cela un dictionnaire **D_traj_opt** dont les clefs sont les coordonnées des positions et les valeurs les trajectoires optimales pour rejoindre ces positions à partir du point D. La première clef de ce dictionnaire est **(0, 0)** et la valeur correspondante **[]**. Soit : **D_traj_opt[(0, 0)] = []**.