

DS (rattrapage) - T NSI

EX 1 (12 points)

On appelle **deque** (double-ended queue ou deque en Anglais) une structure de donnée linéaire à la fois pile et file, ce qui signifie qu' on peut insérer ou supprimer un élément dans la structure de chaque côté (gauche ou droite) de la structure linéaire.

Le but de l'exercice est d'implémenter en Python une classe Deque, comme un ensemble de Noeud, à partir d'une classe Noeud définie par :

```
class Noeud:
    def __init__(self,v,p,s):
        self.valeur = v
        self.precedent = p
        self.suivant = s
```

La classe Deque a deux attributs nommés, gauche et droit.

Voici l'API (Application Programming Interface) que vous devez implémenter :

```
class Deque:
    #construit une deque vide
    def __init__(self):
        pass

    #est ce que la deque est vide?
    def est_vide(self):
        pass

    #ajoute un élément sur la gauche
    def ajoute_gauche(self,element):
        pass

    #ajoute un élément sur la droite
    def ajoute_droit(self,element):
        pass

    #supprime un élément de la gauche et renvoie cet élément
    def supprime_gauche(self):
        pass

    #supprime un élément de la droite et renvoie cet élément
    def supprime_droit(self):
        pass
```

Cas limites

1. Si on veut enlever (à gauche ou à droite) d'une deque vide, il faut lever une exception **IndexError**
2. Si on veut ajouter None à une deque il faut lever une exception **ValueError**

EX 2 (8 points)

Le propriétaire d'une pension pour animaux gère les places dont il dispose à l'aide d'une base de données dont voici le schéma relationnel (sans préciser les clés primaires et étrangères éventuelles)

- client(num_client, nom_client, prenom_client, mail_client, tel_client)
- animal(num_animal, nom_animal, categorie_animal, taille_animal, num_client)
- cage(num_cage, taille_cage, secteur_cage)
- reservation(num_reservation, date_debut, date_fin, num_client, num_animal, num_cage)

On donne ci-dessous des extraits des tables

num_client	nom_client	prenom_client	mail_client	tel_client
16	Euler	Leonhard	leo@bd.ch	0703141592
64	Turing	Alan	alan@bd.uk	0123456789

num_animal	nom_animal	categorie_animal	taille_animal	num_client
2	Bip-Bip	souris	petit	16
64	Balou	chien	grand	16
100	Bagheera	chat	moyen	64

num_cage	taille_cage	secteur_cage
10	petit	rongeur
60	grand	chien
200	moyen	chat

num_reservation	date_debut	date_fin	num_client	num_animal	num_cage
78	2023-12-23	2023-12-26	16	2	10
79	2023-12-23	2023-12-26	16	64	60
80	2023-12-23	2024-1-2	64	100	200

1. Préciser pour la relation reservation, le (ou les attributs) pouvant être utilisé(s) comme clé primaire. Quel est l'intérêt d'une clé primaire ?
2. Préciser pour la relation reservation, le (ou les attributs) pouvant être utilisé(s) comme clé étrangère. Quel est l'intérêt d'une clé étrangère ?
3. Ecrire en SQL une requête permettant d'afficher les noms de tous les chiens hébergés dans la pension.
4. Ecrire en SQL une requête permettant d'afficher les noms et la catégorie des animaux du client dont le nom est 'Euler' et le prénom 'Leonhard'
5. Ecrire en SQL une requête permettant d'insérer un nouveau client dont le nom est 'Dreistein', le prénom est 'Albert', le mail 'albert@bd.de' et le numéro de téléphone '0299999999'. Pour l'instant le plus grand num_client est 216.
6. Il y a 400 cages dans la pension. Ecrire une requête en SQL permettant de savoir si la pension est complète.
7. Le propriétaire aimerait supprimer des clients ainsi que leurs animaux de la base de données.

Dans quel ordre doit-il procéder ? (d'abord supprimer le client puis le ou les animaux du client, ou alors supprimer le ou les animaux du client puis le client). Justifier