Devoir sur table.
P.O.O & Base de donnée

Exercice 1

Pour simplifier la gestion de ses mots de passe, Alice décide de créer un générateur de mots de passe ainsi qu'un gestionnaire de mots de passe regroupant les informations de connexion de chaque site qu'elle utilise.

Partie A

Dans cette partie, on s'intéresse à la création des mots de passe d'Alice. Pour générer ses mots de passe, Alice décide de créer un programme `Pyhton`. Elle a commencé à créer une classe `MDP` lui permettant de créer aléatoirement un mot de passe, d'afficher le mot de passe et tester la résistance de celui-ci.
Le code qu'elle a produit ce trouve ci-dessous :

```
from random import randint

class MDP:

    def __init__(self):
        self.minuscules = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n',
        ↪  'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z']
        self.majuscules = ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N',
        ↪  'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z']
        self.speciaux = ['!', '"', '#', '$', '%', '&', "'", '(', ')', '*', '+', ',', '-', '.', '/',
        ↪  ':', ';', '<', '=', '>', '?', '@']

    def gen_mdp(lon ,maj=True, min=True, spe=True):
        '''
        Méthode pour générer des mot de passe:

        params :
            - lon (int) : longueur du mot de passe
            - maj (bool) : vraie si le mot de passe doit contenir des majuscules
            - min (bool) : vraie si le mot de passe doit contenir des minuscules
            - spe (bool) : vraie si le mot de passe doit contenir des caractères spéciaux
        '''
        assert maj or min or spe, "Le mot de passe doit être composé d'au moins une série de
        ↪  caractères"
        ..........................................................................
        caracteres = []
        if maj:
            # Ajout les lettres majuscules au tableau caracteres
            .caracteres.extend(............)
        if ... :
            # Ajout les lettres minuscules au tableau caracteres
            ..................................
        ........
            # Ajout des caractères spéciaux au tableau caractères
            ..................................
        # création du mot de passe de  la longueur demandée
        for i in range(.....):
            self.mdp += caracteres[randint(.......)]
```

1. Quels sont les attributs de la classe `MDP` ?
2. Parmi les paramètres de la méthode **`generer`** de la classe `MDP`, lesquels sont optionnelles ?
3. La variable **`caracteres`** de la méthode **`generer`** est une liste contenant tous les caractères autorisés pour fabriquer un nouveau mot de passe.

Recopier et compléter les lignes 25, 26, 28, 29 et 31 du code de la fonction `generer`, puis écrire les autres lignes de code afin de créer la variable `caracteres`.

> *i* La fonction `randint` du module `random` permet de générer un nombre aléatoire entre O et un nombre entier entré en paramètre.
>
> Exemple : `randint(5)` : permet de générer un nombre entier pseudo aléatoire compris entre 0 et `5`. (les deux bornes étant inclus)

4. On suppose maintenant que la variable `caracteres` est correctement créer.
   Recopier et complèter les lignes 34 de la méthode `generer`, pour qu'elle génère un mot de passe aléatoire et le stock dans l'attribut `mdp`.
5. Pour afficher le mot de passe, on souhaite ajouter la méthode `__repr__` qui retounre l'attribut `mdp` de la classe `MDP`

   Alice s'inscrit sur un nouveau site lui demandant de créer un mot de passe de 8 caractères minimum, composé uniquement de majuscules et de minuscules.
6. Instancier un objet de la classe `MPD` (nommé `pass`) puis utiliser la méthode pour `generer` pour générer un mot de passe répondant aux caractèristique du site.
7. Expliquer en quoi le mot de passe ainsi générer peut renvoyer un mot de passe qui ne répond pas aux exigences du site.

   On donne ci-dessous deux bonnes pratiques parmi celles proposées par le gouvernement en matière de gestion des mots de passe.

   - **P1. Utilisez un mot de passe différent pour chaque service.**
     Ainsi en cas de perte ou de vol d'un de vos mots de passe, seul le service concerné sera vulnérable. Dans le cas contraire, tous les services pour lesquels vous utilisez le même mot de passe compromis seraient piratables..
   - **P2. Utilisez un mot de passe suffisamment long et complexe.**
     Une technique d'attaque répandue, dite par « force brute », consiste à essayer toutes les combinaisons possibles de caractères jusqu'à trouver le bon mot de passe. Réalisées par des ordinateurs, ces attaques peuvent tester des dizaines de milliers de combinaisons par seconde. Pour empêcher ce type d'attaque, il est admis qu'un bon mot de passe doit comporter au minimum 12 signes mélangeant des majuscules, des minuscules, des chiffres et des caractères spéciaux

   Source : d'après https ://www.cybermalveillance.gouv.fr/tous-nos-contenus/bonnes-pratiques/mots-de-passe
8. Modifier l'assertion en ligne 20 pour quelle prenne en compte les recommendation du gouvernement concernant les caractères à utiliser.
9. Toujours en suivant les recommendations du gouvernement. Créer l'assertion en ligne 21 qui vérifie que la longueur demandé pour le mot de passe et le cas échant lèvent l'assertion `"Mot de passe trop court"`.
10. Expliquer pourquoi même avec ces changements le mot de passe généré peut ne pas suivre les recommendations du gouvernement
11. En se basant sur les recommendations du gouvernement suivantes :
    - la longueur du mot de passe doit être au moins de 12 caractères.
    - le mot de passe comporte au moins un caractère de chaque type (majuscule, minuscules et caractere spécial)

    Ecrire une méthode `est_securise` de la classe `MDP` qui renvoie `True` si l'attribut `mdp` suit les recommendations ci-dessus et `False` sinon.

Partie B
Dans cette partie, on pourra utiliser les instructions du langage SQL pour :

Alice souhaite mettre en œuvre une base de données composée des deux relations compte et site dont des **extraits** sont donnés dans les tableaux suivants.

compte			
mot_de_passe	utilisateur	renouvellemen	id_site
Asrtg!Myfj	aliceB24	2022-06-30	1
@rDfohpj!	aliceB24	2021-03-12	2
GxRGDxc(u-PM	alice_B@votremailp.me	2018-10-14	4
Ghcj=+f*AZs	alice1276	2022-06-30	3
cYFgt!:Ehr;	alice_B2@votremailp.m	2022-06-30	4

site		
id	nom_site	url
1	Vosnotes	https ://logi-educ.net/vosnotes/eleve.html
2	Banque Perso	https ://www.banqueperso.fr/connexion.html
3	Elec verte	https ://espace-client.ev.fr/login
4	Votremailp	https ://account.votremailp.me/login

- L'attribut `mot_de_passe` est une clé primaire de la table compte.
- L'attribut `id` est une clé primaire de la table site.
- L'attribut `renouvellement`, correspondant au dernier renouvellement du mot de passe, est une chaîne de caractères de format `AAAA-MM-JJ`.
  Ainsi un mot de passe renouvelé le 21 février 2025 correspond à un attribut renouvellement de '2025-02-21'.

12. Dans la table `compte`, l'attribut `id_site` est une clé étrangère référençant l'attribut `id` de la table `site`. Expliquer son rôle.
13. Expliquer en quoi il n'est pas possible, pour Alice, d'avoir le même mot de passe pour deux sites différents.
14. Écrire la requête SQL permettant d'afficher, sans doublon, toutes les URL enregistrées dans la base de données.
15. Ecrire une requête SQL qui permet de compter le nombre de comptes utilisés par Alice avec l'identifiant `AliceB24`.
16. La Banque Perso a demandé à Alice de renouveler son mot de passe. Elle remplace le mot de passe `@rDfohpj!` par `yhTS?d@UTJe`.
    Écrire la requête SQL permettant de faire la modification dans la base de données. On ne s'occupera pas de la modification de l'attribut renouvellement.

    On rappelle que le langage SQL utilise l'ordre lexicographique (ordre du dictionnaire) pour comparer deux éléments de type texte. Selon cet ordre, on a par exemple '3' supérieur à '1' et 'python' est inférieur à 'sql'.

17. À la date du 13 novembre 2025, Alice a décidé de lister tous les mots de passe qui n'ont pas été renouvelés depuis plus d'un an.
    Écrire la requête SQL permettant de donner la liste des id_site concernés.
18. Écrire une requête SQL permettant d'afficher tous les utilisateurs et les mots de passe du ou des sites de nom 'Votremailp' dont l'identifiant est supposé non connu. Le résultat devra être affiché par ordre chronologique de date de renouvellement de mot de passe.