Examen de mi-session, Programmation2, INF1093

Directives importantes :

1. *L'examen est surveillé, aucune communication n'est autorisée*
2. *La documentation est autorisée mais exclut le web. Donc une fois l'examen lance, aucun navigateur web ouvert ne sera autorisé.*
3. *Durée : 3h*
4. *Créé un fichier python mi_session.py qui contiendra tous tes scripts attendus en réponse aux questions des exercices 2 et 3*
5. *Ensuite, soumets d'avance ce fichier via votre référentiel git. Plus aucun retard ne sera autorisé. Mieux vaux un fichier vierge qu'aucune soumission*
6. *Pour soumettre, exécute les commandes git suivantes a partir de la racine de ton référentiel git.*
    - *git add .*
    - *git commit -m '' numéro de la question ''*
    - *git push*

Tu devras également faire la même chose après ta réponse à chaque question.

L'enseignant devra faire la ronde pour s'assurer de la bonne soumission des solutions tout au long de l'évaluation

Exercice 1 : 30 points

Considérons une liste de couples suivante :

$list = [(n_1, d_1), (n_2, d_2), \ldots (n_k, d_k)]$ Ou $n_i \; et \; d_i$ sont respectivement les noms et âges des étudiants.

Exemple :

Itteration Num:	**Evolution du de la liste**					
0	(Viny, 34)	(Ryan, 43)	(Tity, 31)	(Antony, 27)	(Calvin, 39)	(Lilian, 27)
1						
2						
3						
4						
5						
6						
7						
8						

Crée un fichier Excel nomme **mi-session** et sauvegarde ce fichier a la racine de ton référentiel.

1- Dans une feuille de ce fichier Excel, nommée **question1**, Illustre ta compréhension du **Tri-bulle** en décrivant, par écrit, sur un format A4, itération après itération, l'évolution de la liste jusqu'à son état trié selon les noms en ordre alphabétique **15 points**
2- Dans une feuille de ce fichier Excel, nommée **question2**, Illustre ta compréhension du **Tri-Sélection** en décrivant itération après itération l'évolution de la liste dans ce tableau jusqu'à son état trié selon les âges, du plus jeune au plus vieux. **15 points**

Exercice 2 : **20 points**

Écris une fonction **build_list()** qui collecte les données utilisateurs selon le déroulement décrit dans la capture de la figure 1 suivante.

i. Après chaque couple de valeur saisie, la fonction demande à l'utilisateur s'il veut encore ajouter un nouvel étudiant. Si oui, l'utilisateur entre 1, sinon, l'utilisateur entre 0
ii. La fonction doit renvoyer la liste bâtie en retour avant qu'une instruction d'affichage permette d'en afficher le contenu. Cette instruction d'affichage ne fait pas partie de la fonction.

```
Nom: Viny
Age: 34
Ajouter un autre etudiant? (1/0)1


Nom: Ryan
Age: 43
Ajouter un autre etudiant? (1/0)1


Nom: Tity
Age: 31
Ajouter un autre etudiant? (1/0)1


Nom: Antony
Age: 27
Ajouter un autre etudiant? (1/0)1


Nom: Calvin
Age: 39
Ajouter un autre etudiant? (1/0)1


Nom: Lilian
Age: 27
Ajouter un autre etudiant? (1/0)0


Liste complete non triee:   [('Viny', 34), ('Ryan', 43), ('Tity', 31), ('Antony', 27), ('Calvin', 39), ('Lilian', 27)]
```

Figure 1 : Collecte des données utilisateurs

Exercice 3 **50 points**

1. Écris une fonction **switch (liste, i, j)** qui permute deux éléments contenus dans une liste passée en paramètre. Les positions des 2 éléments à permuter sont données par les indices **i** et **j** **10 points**

```
Liste complete non triee:   [('Viny', 34), ('Ryan', 43), ('Tity', 31), ('Antony', 27), ('Calvin', 39), ('Lilian', 27)]
Liste complete apres permutation des element 1 et 2:   [('Viny', 34), ('Tity', 31), ('Ryan', 43), ('Antony', 27), ('Calvin', 39), ('Lilian', 27)]
```

Figure 2 Résultat de la permutation entre les éléments 1 et 2

2. Utilise cette fonction de permutation pour implémenter une fonction **bubbleSort(liste)** qui permet de trier par bulle la liste produite dans l'exercice précédent. Ce tri est basé sur les noms et sera en ordre alphabétique

   Afficher le contenu de la liste après chaque itération. Comparer le rendu avec l'évolution décrit dans la question 1 de l'exercice 1 **20 points**

3. Utiliser la même fonction pour implémenter une fonction **selectionSort(liste)** permettant de trier par sélection la même liste. Ce tri est basé les âges et sera en ordre croissant. **20 points**