## Exercice 3 (8 points)

*Cet exercice porte principalement sur les systèmes d'exploitation, les réseaux et la programmation de base en Python.*

Une association de jardinage anime un réseau d'échange de plantes.

### Partie A

Les échanges de plantes sont traités à travers un système de gestion de fichiers dans un espace de stockage partagé.

Le répertoire `association` est positionné à la racine du système de fichiers.

Un répertoire `annonces` contient des fichiers au format `HTML` décrivant l'échange souhaité. Le fichier contient notamment le nom de la personne qui soumet l'annonce et décrit la plante proposée ainsi que le type de plante souhaitée en échange. Chaque adhérent dispose d'un répertoire à son nom à l'intérieur du répertoire `adherents`.

La figure ci-dessous donne un extrait de l'arborescence :

```
|---+ association
    |---+ adherents
    |   |---+ abi
    |   |   |--tulipe_chen.html
    |   |   |--muguet_abi.html
    |   |
    |   |---+ bachir
    |   |   |--reseda_dana.html
    |   |
    |   |  [...]
    |
    |---+ annonces
    |   |--rosier_abi.html
    |   |--pivoine_bachir.html
```

L'utilisateur manipule le système de fichiers à travers un terminal (en ligne de commande). Il travaille sur un système de type Unix.

1. Décrire le résultat de la commande `ls /association` saisie dans un terminal.

2. Le répertoire courant dans le terminal est `/association`. Parmi les commandes systèmes suivantes, identifier celle qui permet de définir `annonces` comme répertoire courant.

- `cd /annonces`
- `cd ../annonces`
- `cd annonces`
- `cd association/annonces`

Les quatre premières lignes de la documentation obtenue avec la commande `cp --help` sont les suivantes :

```
Utilisation : cp [OPTION]... [-T] SOURCE DEST
         ou : cp [OPTION]... SOURCE... RÉPERTOIRE
         ou : cp [OPTION]... -t RÉPERTOIRE SOURCE...
Copier la SOURCE vers DEST ou plusieurs SOURCEs vers
RÉPERTOIRE.
```

Les quatre premières lignes du résultat de `mv --help` sont :

```
Utilisation : mv [OPTION]... [-T] SOURCE DEST
         ou : mv [OPTION]... SOURCE... RÉPERTOIRE
         ou : mv [OPTION]... -t RÉPERTOIRE SOURCE...
Renommer SOURCE en DEST, ou déplacer le ou les SOURCEs vers
RÉPERTOIRE.
```

3. Expliquer la différence entre les commandes suivantes en précisant le nombre d'exemplaires du fichier `rosier_abi.html` à l'issue de chacune d'elles.

   - `cp rosier_abi.html ../adherents/abi/`
   - `mv rosier_abi.html ../adherents/abi/`

Abi propose une annonce à laquelle Bachir voudra répondre. L'annonce d'Abi figure dans la page `rosier_abi.html` qui s'affiche ainsi dans un navigateur :

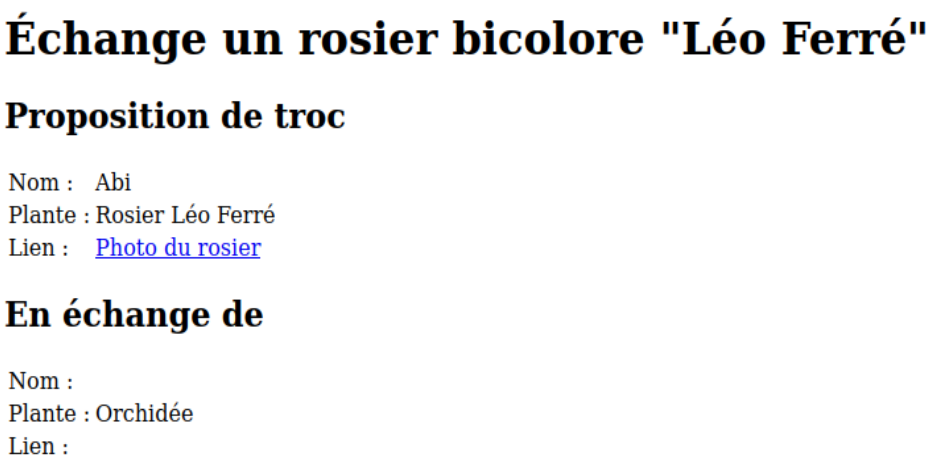



Figure 1. rosier_abi.html

Bachir devra répondre à l'annonce dans le code `HTML` de la page `rosier_abi.html` ci-dessous.

```
<html><body>
<h1>Échange un rosier bicolore "Léo Ferré"</h1>

<h2>Proposition de troc</h2>
<table>
    <tr><td>Nom :</td> <td>Abi</td></tr>
    <tr><td>Plante :</td><td>Rosier Léo Ferré</td></tr>
    <tr><td>Lien : </td><td><a
href="monsiteperso.fr/abi/photo_rosier.jpg">Photo du
rosier</a></td></tr>
</table>

<h2>En échange de</h2>
<table>
    <tr><td>Nom :</td><td></td></tr>
    <tr><td>Plante :</td><td>Orchidée</td></tr>
    <tr><td>Lien : </td><td></td></tr>
</table>
</body></html>
```

Pour répondre à l'annonce Bachir doit écrire au bon endroit dans le code source de la page `HTML` les informations suivantes :

- Nom : Bachir
- Plante : Orchidée noire Cymbidium
- Lien : `monsiteperso.fr/bachir/orchidee.jpg`

4. Recopier et compléter la section de code que Bachir doit modifier pour répondre à l'annonce.

Lorsqu'une personne veut répondre à l'annonce, elle procède selon l'algorithme suivant :

- Étape 1 : elle ouvre la page `HTML` dans un éditeur ;
- Étape 2 : elle ajoute les informations nécessaires à l'échange ;
- Étape 3 : elle enregistre les modifications ;
- Étape 4 : elle fait une copie de l'annonce complétée vers son répertoire personnel ;
- Étape 5 : l'exemplaire original modifié est déplacé vers le répertoire personnel de la personne qui a proposé l'annonce.

À l'issue de cet algorithme, l'annonce a disparu du répertoire `annonces`.

5. Bachir et Chen veulent tous les deux répondre à l'annonce. Ils exécutent chacun l'algorithme sensiblement à la même heure.

   En détaillant étape par étape un exemple, expliquer pourquoi cet algorithme se comporte mal avec la mise en concurrence des deux propositions d'échange.

6. Proposer un nouvel algorithme qui empêche le conflit précédent.

## Partie B

Les membres de l'association s'organisent en réseau afin de s'échanger les plantes de la main à la main.

Certains membres de l'association se rencontrent très régulièrement. Dans ce cas, on dira qu'ils sont amis. Quand un membre de l'association doit faire parvenir une plante, il la confie à un ami qui lui-même la confiera à quelqu'un d'autre, à l'image des routeurs au cœur d'Internet qui se transmettent les messages à router.

Pour acheminer au mieux les plantes, l'association s'inspire du protocole RIP. Ce protocole de routage s'adapte aux modifications du réseau.

On rappelle que le protocole RIP vise à minimiser le nombre de sauts sur les chemins de routage construits. Dans le contexte de l'exercice, il s'agira de minimiser le nombre d'échanges entre adhérents pour faire parvenir la plante à son destinataire.

Les membres sont donc amenés à servir d'intermédiaire. Dans ce cadre, chaque membre construit l'équivalent d'une table de routage.

La table de routage d'Abi est la suivante :

| Table de routage d'Abi | | |
|---|---|---|
| Destinataire | Intermédiaire | Distance |
| Bachir | Bachir | 1 |
| Chen | Bachir | 2 |
| Dana | Dana | 1 |
| Edie | Dana | 2 |

D'après cette table, Abi est amie avec Bachir. Si Abi doit faire parvenir une plante à Chen, elle doit la confier à Bachir car il est un ami de Chen. La distance correspond au nombre de rencontres nécessaires pour faire parvenir la plante.

*Exemple : Dans le tableau, on lit que la distance entre Abi et Chen vaut 2 car si Abi veut faire parvenir un rosier à Chen, il faut que Abi confie le rosier à Bachir puis que Bachir le donne à Chen. Edie n'est pas un ami d'Abi, son nom ne peut pas apparaitre dans la colonne « Intermédiaire ».*

7. Reproduire le graphe ci-dessous puis ajouter une arête pour que le graphe devienne cohérent avec la table de routage d'Abi.

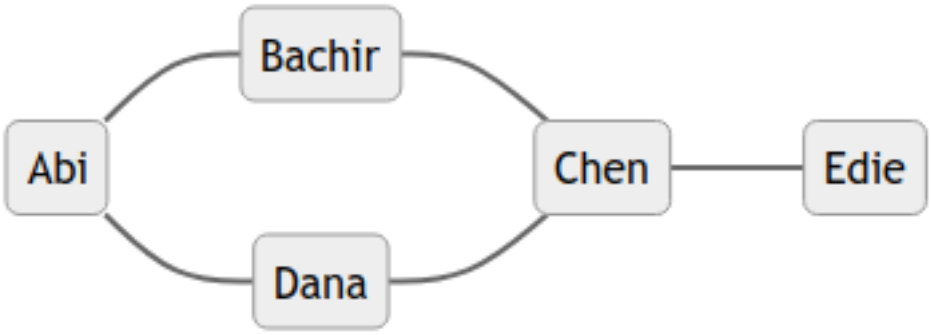



Figure 2. Relations entre les membres de l'association

Frida rejoint l'association. Elle est amie avec Abi. À son arrivée dans l'association, Frida ne connaît personne d'autre. Pour construire sa table de routage, Frida exploite les informations de la table de routage d'Abi.

8. Reproduire et compléter la table de routage de Frida.

| Table de routage de Frida | | |
|---|---|---|
| Destinataire | Intermédiaire | Distance |
| Abi | Abi | 1 |
| Bachir | | |
| Chen | | |
| Dana | | |
| Edie | | |

Abi met également sa table de routage à jour pour y inclure Frida, puis elle va communiquer sa table de routage à tous ses autres amis (Bachir et Dana). Par la suite, Bachir et Dana, à leur tour, vont communiquer leur table de routage à leurs amis, et ainsi de suite.

9. Décrire les modifications que doivent faire Abi, Bachir, Chen, Dana et Edie dans leur table de routage respective suite à l'arrivée de Frida.

Après quelques semaines les relations ont évolué, certains liens se sont rompus et un nouvel adhérent est arrivé. Abi et Frida échangent les informations de leur table de routage respective afin de les actualiser.

| Table de routage d'Abi | | |
|---|---|---|
| Destinataire | Intermédiaire | Distance |
| Bachir | Bachir | 1 |
| Chen | Bachir | 3 |
| Dana | Dana | 1 |
| Edie | Dana | 2 |
| Frida | Frida | 1 |
| Guy | Dana | 3 |
| Hakim | Hakim | 1 |

| Table de routage de Frida | | |
|---|---|---|
| Destinataire | Intermédiaire | Distance |
| Abi | Abi | 1 |
| Bachir | Abi | 2 |
| Chen | Chen | 1 |
| Dana | Abi | 2 |
| Edie | Guy | 2 |
| Guy | Guy | 1 |

10. Décrire les modifications que chacune d'elles doit apporter à sa table de routage afin de maintenir les routes les plus courtes pour chaque destinataire.

## Partie C

Dans cette partie, les tables de routage sont structurées sous forme de dictionnaires. Les clés du dictionnaire sont les destinataires. La valeur associée à un destinataire est le tuple `(intermediaire, distance)`.

La table de routage d'Abi est la suivante :

```
1 table_abi = {'Abi'      : ('Abi'    , 0),
2              'Bachir'   : ('Bachir' , 1),
```

```
3              'Chen'     : ('Bachir' , 3),
4              'Dana'     : ('Dana'   , 1),
5              'Edie'     : ('Dana'   , 2),
6              'Guy'      : ('Dana'   , 3),
7              'Hakim'    : ('Hakim'  , 1)}
```

La table de routage d'Hakim est la suivante :

```
1 table_hakim = {'Ines'    : ('Janus'  , 2),
2                'Janus'   : ('Janus'  , 1)}
```

11. En vous appuyant sur la table de routage d'Hakim, répondre par vrai ou faux à chacune des affirmations suivantes :
    - Hakim et Janus sont amis ;
    - Hakim et Ines sont amis ;
    - Janus et Ines sont amis.

La fonction `amis` prend en argument une table de routage d'une personne et elle renvoie la liste de ses amis, c'est-à-dire la liste des intermédiaires, sans doublon. La fonction a été correctement programmée, mais par erreur les lignes de codes ont été mélangées (elles ont été triées dans l'ordre alphanumérique croissant). Les espaces en début de ligne ont été conservés et ils sont donc corrects, seul l'ordre des lignes a été modifié.

```
1             liste.append(intermediaire)
2         if intermediaire not in liste:
3     """renvoie la liste des intermédiaires de la table de
routage, sans doublon"""
4     for (intermediaire, distante) in table.values():
5     liste = []
6     return liste
7 def amis(table):
```

12. Remettre les lignes de codes dans le bon ordre.

Dans le but d'automatiser la mise à jour des tables de routages, il est proposé le programme suivant :

```
1 def maj(ma_table, ami, table_ami):
2     """mise à jour de ma_table (dict) avec les
3     informations de table_ami (dict).
4     ami est du type str."""
5
6     if ami not in ma_table.keys():
7         ma_table[ami] = ( ... , ... )
8
```

```
 9     for adh in table_ami.keys():
10         (intermediaire, distance)  = table_ami[adh]
11         if adh not in ma_table.keys():
12             ma_table[adh] = (intermediaire, distance + 1)
13         if ma_table[adh][...] > distance + 1:
14             ma_table[adh] = (ami, ...)
15
16     for adh in ma_table.keys():
17         (intermediaire, distance) = ma_table[adh]
18         if adh not in table_ami.keys() and adh != ami and
intermediaire == ami:
19             ma_table[adh] = (None, float('inf'))
20
21     return ma_table
```

La fonction `maj` prends en paramètres :

- un dictionnaire `ma_table` représentant une table de routage ;
- une chaîne de caractères `ami` désignant le nom de l'ami ;
- un dictionnaire `table_ami` représentant la table de routage de l'ami. C'est de cette table de routage que sont extraites les informations utiles.

Cette fonction met à jour la table de routage `ma_table`.

13. Expliquer le test de la ligne 6 du code de la fonction `maj`.
14. Recopier et compléter la ligne 7 du code de la fonction `maj`.
15. La boucle de la ligne 9 parcourt les clés de `table_ami` pour mettre à jour les informations de `ma_table`. Recopier et compléter les lignes 13 et 14 du code de la fonction `maj`.

La fonction `nettoie`, dont le code est donné ci-après, prend en paramètre un dictionnaire `table` représentant une table de routage. Elle a pour objectif de supprimer les entrées du dictionnaire des membres devenus injoignables (suite à la mise-à-jour de la table dans la fonction `maj` aux lignes 16 à 19).

```
1 def nettoie(table):
2     """Supprime toute les noms qui ne sont pas joignables"""
3     for (adh, ligne) in table.items():
4         if ligne[0] == None:
5             del table[adh]
```

Il est impossible, avec le langage Python, de supprimer des entrées dans un dictionnaire à l'intérieur d'une boucle qui le parcourt. C'est la raison pour laquelle, à l'exécution de cette fonction avec une certaine table, on obtient l'erreur suivante :

```
Traceback [...]line 1, in nettoie
    for (adh, ligne) in table.items():
RuntimeError: dictionary changed size during iteration
```

16. Proposer une version corrigée de la fonction `nettoie` qui évite le déclenchement de l'erreur décrite.