

Initiation à la programmation – TP15

25 novembre

Objectif : Fichiers textes en python

Exercice – travail sur un annuaire téléphonique

On travaille sur des fichiers textes qui sont des annuaires téléphoniques composés de la manière suivante : chaque ligne contient un nom, un prénom et un numéro à dix chiffres (écrit sans espaces).

Par exemple le fichier `bottinNomPrenom.txt` (que vous pouvez télécharger) contient :

```
dupont pierre 0657677834
dumas leon 0212100339
marie lucie 0757673423
martin luc 0567342312
monnier annette 0231562389
albert lucie 0222321212
lelong annette 0645121212
marc jean 0456907802
dupont remi 0231567474
durand jacques 0231563435
```

1. (a) Ecrire la fonction `recherche(nom, prenom, fichier)` qui renvoie le numéro correspondant à `nom`, `prenom` dans le bottin qui est dans `fichier` et `""` si aucune ligne ne convient.

```
>>> recherche("dupont","eric","bottinNomPrenom.txt")
""
>>> recherche("dupont","pierre","bottinNomPrenom.txt")
'0657677834'
```

- (b) Ecrire la fonction `rechercheNom(nom, fichier)` qui renvoie la liste des lignes (sans le caractère de saut de ligne) correspondant à `nom` dans le bottin qui est dans `fichier` et `[]` si aucune ligne ne convient.

```
>>> rechercheNom("dupont","bottinNomPrenom.txt")
['dupont pierre 0657677834', 'dupont remi 0231567474']
>>> rechercheNom("dupontel","bottinNomPrenom.txt")
[]
```

- (c) Ecrire la fonction `combien(pre,fichier)` qui renvoie le nombre de numéros de téléphone du bottin contenu dans le fichier `fichier` qui commence par le préfixe `pre`. Remarque : le préfixe `pre` est une chaîne de caractères de longueur quelconque.

```
>>> combien("02","bottinNomPrenom.txt")
5
```

2. On veut faire une recherche améliorée par rapport à un annuaire inversé : on dispose d'un numéro qui peut être incomplet dans lequel les chiffres qu'on ne connaît pas sont remplacés par des étoiles.

- (a) Ecrire la fonction `ressemble(num, ref)` qui teste si le numéro `num` qui est une chaîne de caractères contenant 10 caractères exactement dont éventuellement des étoiles peut correspondre au numéro `ref` qui est une chaîne de caractères contenant 10 caractères chiffres exactement.

```
>>> ressemble("0231**4556","30231674556")
False
>>> ressemble("0231**4556","0231674556")
True
>>> ressemble("0231**4556","0231687556")
False
```

- (b) En déduire la fonction `compatible(ref, bottin)` qui va afficher successivement les lignes dont les numéros sont compatibles avec `ref` (on utilisera la fonction précédente).

```
>>> compatible("*****12","bottinNomPrenom.txt")
martin luc 0567342312
albert lucie 0723121212
lelong annette 0645121212
```

3. Ecrire la fonction `interneFac(fichier)` qui étant donné un fichier qui contient un bottin, provoque la création d'un fichier `Facfichier` (on rajoute la chaîne « Fac » devant le nom du fichier) dans lequel seront inscrits tous les abonnés dont le numéro commence par 02 31 56. Dans ce fichier on ne mettra alors que le nom et les quatre derniers chiffres (correspondant à un numéro interne de l'UCNB). Pour l'exemple ci-dessus le fichier comprendrait donc :

```
>>> compatible("*****12","bottinNomPrenom.txt")
monnier annette 2389
dupont remi 7474
durand jacques 3435
```

4. (a) Ecrire la fonction `ajoute0(nom, prenom, num, fichier)` qui étant donné un nom, un prenom, un numéro (sous la forme '0231565667') et un bottin contenu dans le fichier, ajoute à la suite du fichier le nom, prénom et le téléphone (première version sans test).
- (b) Ecrire la fonction `ajoute(nom, prenom, num, fichier)` qui étant donné un nom, un prenom, un numéro (sous la forme '0231565667') et un bottin contenu dans le fichier, ajoute à la suite du fichier le nom, prénom et le téléphone si l'utilisateur n'est pas déjà présent. S'il y est déjà, on doit prévenir l'utilisateur en indiquant : dupont était dans le bottin avec le 0231566867 est-ce que je le remplace par 0231565667? Si l'utilisateur répond oui, la modification est faite sinon il n'y aura pas d'ajout.
5. Deux amis comparent leurs annuaires personnels. Ecrire la fonction `commun(bot1,bot2)` qui renvoie la liste des t-uples (nom,prénom) des personnes qui sont dans les 2 bottins avec le même numéro de téléphone.
6. On veut aussi pouvoir gérer le bottin avec un dictionnaire : les clés seront formées par les tuples (nom,prénom) et les valeurs seront les numéros de téléphone.

```
>>> dico
{('marc', 'jean'): '0456907802', ('durand', 'jacques'): '0231563435'...}
```

- (a) Ecrire la fonction `creationDico(fichier)` qui crée un tel dictionnaire avec les données du fichier.
- (b) Ecrire la fonction `sauvegarde(dico, f)` qui va écrire les données du dictionnaire `dico` dans le fichier `f` selon le format des premières questions (ce fichier sera remis à zéro auparavant).