Instruction d'utilisation d'annuaire:

```
1. Exécuter 'serveur.py' (Le serveur attends des connexions entrantes)
2. Exécuter 'annuaire.py'
3. Se connecter avec un identifiant existant
```

Attention, pour se connecter à l'interface d'administration, il faut utiliser le compte admin

```
    Identifiant : admin
    Mot de passe : admin
```

Une fois sur le compte admin, il suffit de créer un compte utilisateur afin de pouvoir se connecter à son interface dédiée.

Plusieurs connexions simultanées

Dans ce programme nous avons décidé de mettre en place du mutli threading afin d'avoir un serveur qui attend les requêtes.
Celui-ci se contente d'écouter en boucle toutes connexion entrantes et de faire le lien via une ip (localhost) et un port définie.

Nous avons ensuite la partie 'Client/Admin' qui permet d'utiliser notre menu selon le type de compte crée

Structure de données

Nous avons choisi d'utiliser un dictionnaire comme structure de données afin de faciliter la gestion des contacts aussi bien pour l'administrateur que pour le développeur qui pourra reprendre les sources du programme.

Ainsi il sera plus simple de rajouter des informations pour un contact comme son nom, son prénom, son adresse, etc.. puisque ces valeurs sont automatiquement associées à sa clé. En cas de modification des sources de l'annuaire, il suffira d'identifier les nouvelles valeurs grâce à la méthode 'dict.items()' et les associer à une variable personnalisée.

Le programme correspond aux attentes demandées cependant plusieurs améliorations pourrait être bénéfiques tel qu'un hashage différent du MD5.
En effet, l'algorithme MD5 n'est aujourd'hui plus considéré comme sur, il est ammenné à disparaitre par d'autre algorithme tel que SHA-3 par exemple qui est plus récente.
L'autre possibilité serait de hashé les mots de passe avec des salts.

L'autre structure de données qui avait été envisagés était d'utiliser SQLlite comme structure de données, celle-ci est plus ergonomique et simple d'utiliation, de plus la fonction de multi threading est native à SQLlite. Elle nécéssitait cependant de la mettre en place et de la préconfiguré.
La dernière possibilité mais qui ne respectait pas la consigne de base aurait été d'utilisé un framework qui reprend toutes les fonctionnalité python comme **Django**

Enfin, nous avons fait le choix d'utiliser le module 'pickle' afin d'importer et exporter le contenu du dictionnaire depuis un fichier.
'*.pkl', ce fichier permet de conserver une sauvegarde du dictionnaire et son format de données binaire empêche son exploitation par un simple éditeur de texte.