Python TP3 : Exceptions et chiffrement

Exceptions

Les exceptions permettent de gérer les erreurs qui peuvent survenir sans « planter » le programme.

https://openclassrooms.com/courses/apprenez-a-programmer-en-python/les-exceptions-4

Hashage

Le hashage permet de stocker et vérifier un mot de passe sans le garder en clair dans un fichier.

```
import hashlib

password = 'mot de passe'
mdp = hashlib.sha512( password.encode() ).hexdigest()
```

Des versions plus sûres existent : sha3, Blake2, BCrypt…

Sinon possibilité de retrouver les mots de passe courants (crackstation.net).

```
import bcrypt
import random

password = "mon mot de passe"
print bcrypt.hashpw(password,bcrypt.gensalt())
```

Travail du TP

1. Reprenez vos programmes du TP1 (menu) et TP2 (calculatrice) et ajouter la gestion des exceptions (try, except, else, finally).

2. Lever une exception avec assert et raise.

3. Demander pour enregistrement un login (input) et un mot de passe à l'utilisateur avec getpass() en mode console ou avec une fenêtre tkinter affichant des * à la place des caractères saisis.
   Stocker le couple login et hash mot de passe dans un fichier texte.
   Demander pour vérification un login et un mot de passe, vérifier la présence dans le fichier.
   Utiliser le « salage » pour renforcer le système avec une chaîne comprenant le login et une partie fixe.

4. Demander à l'utilisateur le nom d'un fichier existant (par ex. un fichier texte que vous avez créé) que vous allez chiffrer dans un nouveau fichier en utilisant le mot de passe précédent avec l'algorithme AES 256. Vous pouvez utiliser la bibliothèque PyCryptodome http://pycryptodome.readthedocs.io/en/latest/ ou une autre si elle pose problème.
   Proposer aussi le déchiffrement du fichier.