Attaque par force brute

L'activité consiste à implémenter un logiciel qui :

- **détermine les mots de passe en utilisant une attaque par force brute ;**
- **qui génère automatiquement des mots de passe forts.**

1. Attaque par force brute

L'attaque par **force brute** est une méthode qui consiste à tester, une par une, toutes les combinaisons possibles.

2. Implémentation

2.1. Détermination des mots de passe

Les mots de passe à trouver devront être soit :

- Une **suite de 6 chiffres**, comme par exemple 123456 ou 000100 ;
- Une **date de naissance valide** au format **`jjmmaaaa`** supérieure à 1 900 ;
- Un **mot de la langue française** (fichier dictionnaire.txt) ;
- Un **mot de passe couramment utilisé** (fichier disponible sur ce site).

2.2. Générateur de mots de passe

Le générateur de mots de passe doit permettre de générer des mots de passe aléatoires résistant à une attaque par force brute. L'utilisateur pourra préciser les caractères à utiliser (chiffres, majuscules, minuscules, caractères spéciaux) et la longueur du mot de passe souhaité.

3. Travail

Travail en binôme ou seul.

Le logiciel créé devra trouver les mots de passe d'un fichier **`pdf`**.
Pour ceci, on utilisera dans ce projet le module **`pymupdf`** de Python permettant d'interagir avec des fichiers au format **`pdf`**.

✍ Implémenter un logiciel qui détermine les mots de passe par force brute et qui propose un générateur de mots de passe comme défini dans le partie Implémentation (ci-dessus).

✍ Réaliser une interface avec le module Tkinter.

✍ Fournir un fichier readme.txt.

✍ Fournir un compte-rendu.

✍ Fournir des fichiers de test.

4. Aide

4.1. Suite de 6 chiffres

Le programme **force_brute_eleve.py** ci-dessous permet de tester le mot de passe **123456** sur le fichier **fichier_123456.pdf**.

```
import fitz     # importer le module pymupdf pour utiliser la bibiothèque fitz

# création d'un objet document à partir du fichier fichier_123456.pdf
doc_pdf = fitz.Document("fichier_123456.pdf")


def test_password(password, doc):
    '''Renvoie True lorsque password permet d'ouvrir le fichier associer à l'objet doc, sinon renvoie False'''
    return doc.authenticate(password) != 0


if test_password("123456", doc_pdf):
    print("Mot de passe trouvé")
else:
    print("Mot de passe non trouvé")

doc_pdf.close()  # fermeture du fichier
```

4.2. Date de naissance valide

Le module **datetime** permet de manipuler des dates.

4.3. Mot issu d'un fichier txt

Le script suivant permet de mettre sous forme de liste les mots contenus dans un fichier txt (ici **dictionnaire.txt**).

```
# ouvre le fichier dictionnaire en lecture avec comme nom d
with open('dictionnaire.txt', 'r') as d:
    liste_dico = d.readlines()  # retourne une liste avec les mots du dictionnaire
```

Attention la liste **liste_dico** créée n'est pas directement exploitable car les caractères accentués (**à** comme ci-dessous par exemple) ne sont pas pris en compte et l'instruction **readlines** a recopiée la ligne entière (le mot de la ligne plus le caractère de saut de ligne : Entrée qui se symbolise par **\n**).

```
>>> liste_dico
['a\n', 'Ã\xa0\n', 'abaissa\n', 'abaissable\n', 'abaissables\n', 'abai …
```

4.4. Générateur de mot de passe

Les bibliothèques **random** et **string** sont à utiliser.