

# NSI - TP sur les séquences: force d'un mot de passe

*Prérequis: parcours d'une chaîne de caractères avec une boucle bornée, fonction occurrence*

*Rappels: On rappelle que les chaînes de caractère possèdent certaines méthodes:*

- *c.islower() : teste si le caractère c est en minuscules.*
- *c.isupper() : teste si le caractère c est en majuscules.*
- *c.isalpha() : teste si le caractère c est une lettre*
- *c.isdigit() : teste si le caractère c est un chiffre*

Un administrateur d'un site web veut assurer un maximum de sécurité pour les utilisateurs du site. Pour ceci il décide de réaliser une application qui évalue la force des mots de passe des différents utilisateurs du site, sachant qu'un mot de passe est une chaîne de caractères qui ne comporte pas d'espaces et de lettres accentuées.

La force d'un mot de passe varie, selon la valeur d'un score calculé, de 'Très faible' jusqu'à 'Très fort' :

- Si le score <20, la force du mot de passe est 'Très faible'
- Sinon si le score <40, la force d'un mot de passe est 'Faible'
- Sinon si le score <80, la force du mot de passe est 'Fort'
- Sinon la force du mot de passe est 'Très fort'

Le score se calcule en additionnant des bonus et en retranchant d'éventuelles pénalités (niveau difficile).

---

**Les bonus attribués sont :**

- Nombre total de caractères \* 4
- (Nombre total de caractères – nombre de lettres majuscules) \* 2
- (Nombre total de caractères – nombre de lettres minuscules) \* 3
- Nombre de caractères spéciaux, parmi (@, &, \_, -, ?, /, +) \* 5

*Exemple*

*Pour le mot de passe 'P@cSI\_promo2017', le score se calcule comme suit :*

- *La somme de bonus =  $15*4 + (15-3) *2 + (15-6) *3 + 2*5 = 121$  • Le nombre total de caractères = 15*
- *Le nombre de lettres majuscules = 3*
- *Le nombre de lettres minuscules = 6*
- *Le nombre de caractères non alphabétiques = 6*
- *Le score final = 121 ; puisque  $121 > 80$  alors le mot de passe est considéré comme 'Très fort'*

---

## Travail demandé - partie 1

1. Ecrire une fonction NbCMin(p) qui retourne le nombre de caractères minuscules.
2. Ecrire une fonction NbCMaj(p) qui retourne le nombre de caractères majuscules.
3. Ecrire une fonction NbCSpec(p) qui retourne le nombre de caractères spéciaux.
4. Ecrire une fonction Score(p) qui affiche le score d'un mot de passe

---

## Pour aller plus loin: Calcul des pénalités

On soustrait au score des pénalités calculées comme ceci:

- La longueur de la plus longue séquence de lettres minuscules \* 2
- La longueur de la plus longue séquence de lettres majuscules \* 3

*Exemple: Pour le mot de passe 'P@cSI\_promo2017'*

- *La longueur de la plus longue séquence de lettres minuscules('promo') =5*
- *La longueur de la plus longue séquence de lettres majuscules('SI') =2*
- *La somme des pénalités = 5\*2 + 2\*3 = 16*

*Le score du mot de passe devient alors: 121 - 16 = 105, ce qui reste 'Très Fort'*

---

## Travail demandé - partie 2

1. Ecrire une fonction LongMaj(p) retourne la longueur de la plus longue séquence de lettres majuscules.
2. Ecrire une fonction LongMin(p) retourne la longueur de la plus longue séquence de lettres minuscules.
3. Modifier la fonction Score(p) qui affichera le score d'un mot de passe, compte tenu des pénalités.

---

## Travail demandé - partie 3

Vous allez maintenant créer une interface pour le client lui demandant s'il veut se connecter à son compte ou créer un mot de passe. Lorsque le mot de passe proposé n'est pas assez fort, affichez lui une information. Et lorsque le mot de passe est assez fort, stockez le dans une liste:

```
L = [(identifiant1, mot_de_passe1), (identifiant2, mot_de_passe2), ...]
```

Il ne pourra y avoir qu'un seul mot de passe par identifiant.

Vous pourrez alors terminer votre projet avec la connexion au compte client lorsque le mot de passe renseigné est correct.

Vous aurez besoin d'ajouter une fonction recherche dans votre programme. Cette fonction retournera l'index de l'élément de liste en faisant une recherche par l'identifiant du compte.

Exemple:

```
> L = [('Gerard Bouchard', 'monSuperMot2Passe@1234'), ('John Doe',  
'BoddySattva456&&')]  
> recherche(L, 'Gerard Bouchard')  
0
```