

DS 4

Informatique pour tous, première année

Julien REICHERT

Dans le premier exercice, nous allons manipuler des chaînes de caractères. On signale ou on rappelle qu'en Python, `s.lower()` donne la chaîne `s` dont tous les caractères correspondant à des lettres sont en bas de casse (c'est-à-dire minuscules) et `s.upper()` donne la chaîne `s` dont tous les caractères correspondant à des lettres sont en capitales (c'est-à-dire majuscules).

Exercice 1a : Quel est le problème du code ci-après ? Écrire une fonction correcte qui prend en entrée une chaîne de caractères et qui renvoie la même chaîne mais dont le premier et le dernier élément sont tous deux remplacés par un guillemet à l'anglaise ("`>`").

```
def remplace(s):  
    s[0] = "<"  
    s[-1] = ">"
```

Exercice 1b : Écrire une fonction qui détermine si une chaîne de caractères est en bas de casse sauf le premier caractère qui est en capitales. S'il existe des caractères ne correspondant pas à des lettres, on peut les ignorer (mais c'est uniquement le premier élément de la chaîne qui doit être en capitales).

Dans le deuxième exercice, nous allons faire un calcul de dénombrement. On considère un ensemble E à n éléments (n sera fixé) et un ensemble F à k éléments (k sera variable).

Exercice 2a : En notant $u_k^{(n)}$ le nombre de surjections de E dans F dans le cas général, montrer la relation de récurrence suivante :

$$u_k^{(n)} = k^n - \sum_{i=0}^{k-1} \binom{k}{i} u_i^{(n)}.$$

Exercice 2b : Écrire une fonction en Python qui calcule $u_k^{(n)}$ avec n et k en entrée. On créera un tableau des $u_i^{(n)}$ pour i de 0 à k qu'on remplira pour les valeurs croissantes de i .

Dans le troisième exercice, nous allons écrire une variante de la méthode d'Euler un peu plus efficace que la version standard. La méthode en question pourrait être qualifiée de semi-explicite, mais il ne s'agit pas de la méthode semi-explicite officielle, qui semble après brève recherche plus complexe et non appliquée à des équations différentielles d'ordre 1 avec une seule fonction à chercher.

Le principe est le suivant : l'approximation retenue dans $y(x_{i+1}) = y(x_i) + \int_{x_i}^{x_{i+1}} G(y(t), t) dt$ de $G(y(t), t)$ n'est pas $G(y(x_i), x_i)$ mais la valeur $G(y(x_{i+1}), x_{i+1})$ qui est calculée d'abord par une méthode d'Euler explicite.

Exercice 3 : Écrire une fonction qui implémente cette méthode, avec pour arguments une fonction `G` décrivant une équation différentielle, une condition initiale `y0` et une liste d'abscisses `X`.

Dans le quatrième exercice, nous allons étudier une base de données. La structure associée est la suivante :

- Table Etudiants, avec les attributs Id (clé primaire avec auto-incrémentation, entier naturel), Classe (chaîne de caractères), Nom (idem) et Prenom (idem). On peut considérer que (Nom, Prénom) est également une clé.
- Table Examens, avec les attributs Id (clé primaire avec auto-incrémentation, entier naturel), Date (chaîne de caractères au format `aaaa/mm/jj`) et Matiere (chaîne de caractères).
- Table Notes, avec les attributs Etudiant (entier naturel), Examen (entier naturel) et Note (entier naturel). Le couple (Etudiant, Examen) est une clé.

Exercice : Écrire une requête SQL permettant de déterminer...

- ... le nombre d'examens d'informatique [question 4a].
- ... la date du premier examen [question 4b].
- ... le nombre de matières où un examen a été enregistré [question 4c].
- ... le nombre de notes au-dessus de dix sur l'ensemble des examens passés en MPSI 2 [question 4d].
- ... la moyenne en informatique de Théo Dupont [question 4e].
- ... le nom de l'étudiant ayant la meilleure moyenne brute (on ne tient pas compte des matières, donc) [question 4f].