

Evaluation Python avancé

Exercice 1 (5 pts)

On va chercher dans cet exercice à écrire des fonctions permettant de corriger une chaîne de caractères. Attention, vous ne devez pas utiliser les fonctions ou méthodes prédéfinies, et vous devez donc construire les nouvelles chaînes à l'aide de boucles.

1. On suppose que la chaîne de caractères a été tapée sur un clavier défectueux, où les lettres 'a' et 'q' sont inversées.

Écrivez une fonction `echange` prenant en argument une chaîne de caractères `chaine` et renvoyant la chaîne de caractères obtenue en remplaçant les occurrences du caractère 'a' par le caractère 'q', et réciproquement, les occurrences de 'q' par 'a'.

2. On suppose maintenant que la personne qui a tapé le texte a une certaine dyslexie, et qu'à chaque fois qu'elle doit taper la séquence 'ea', elle a tapé 'ae' à la place.

Écrivez une fonction `corrige_dys` prenant en argument une chaîne de caractères `chaine` et renvoyant la chaîne de caractères obtenue en corrigeant les défauts présentés précédemment. Plus précisément, à chaque fois que la chaîne contient 'ae', on remplace ces deux caractères par 'ea'.

3. Pour finir, on suppose que la personne qui a saisi le texte a la main un peu lourde, et qu'il lui arrive d'appuyer trop longtemps sur une touche. Or, en français, une lettre n'est jamais répétée plus de deux fois.

Écrivez une fonction `corrige_rep` prenant en argument une chaîne de caractères `chaine` et renvoyant la chaîne de caractères obtenue en corrigeant les défauts présentés précédemment. Plus précisément, à chaque fois qu'un caractère apparaît plus de deux fois consécutivement, alors on élimine les répétitions pour en laisser exactement deux.

Exercice 2 (5 pts)

On considère la fonction f dépendant d'une variable $x \in \mathbb{Z}$ définie par :

$$\forall x \in \mathbb{Z}, f(x) = \begin{cases} x^2 - 35x + 1 & \text{si } x \text{ est pair et } x > 35 \\ -x^2 + 3x + 2 & \text{si } x \text{ est impair et } x \leq 11 \\ x^3 - x^2 + x - 1 & \text{sinon.} \end{cases}$$

On considère ensuite la suite $(u_n)_{n \in \mathbb{N}}$ définie par $u_0 = 10$ et $u_{n+1} = f(u_n)$ pour tout $n \in \mathbb{N}$.

1. Ecrivez une fonction `fonction f` qui prend en entrée un entier `x` et retourne la valeur de $f(x)$.
2. En utilisant cette fonction, écrivez un programme qui demande à l'utilisateur de saisir un entier `x`, puis qui affiche le message suivant : "La valeur de la fonction `f` pour `x=35` est 41684"
3. En utilisant la fonction définie à la question 1, écrivez une fonction récursive `suite u` prenant en entrée un entier naturel `n` et retournant la valeur de u_n .
4. Ecrivez une fonction non récursive `indice_max` prenant en entrée un entier naturel `n`, et retournant l'indice du plus grand des termes $u_0, u_1, u_2, \dots, u_n$.

Exercice 3 (5 pts)

Vous allez écrire des fonctions manipulant des listes d'entiers relatifs (vous ne vérifierez pas que c'est effectivement le cas). Attention, vous ne devez pas utiliser les fonctions prédéfinies pour la recherche d'éléments dans une liste.

1. Ecrire une fonction present qui prend en argument une liste L et un entier x et qui retourne True si x est présent dans L, et False sinon.
2. Ecrire une fonction compte_sup qui prend en argument une liste L et un entier x et qui retourne le nombre d'éléments de L qui sont supérieurs ou égaux à x. Ecrire également une fonction compte_inf qui prend en argument L et x et qui retourne le nombre d'éléments de L qui sont inférieurs ou égaux à x.
3. Un élément x d'un ensemble de N valeurs est appelé une médiane si x vérifie les deux propriétés suivantes :
 - le nombre d'éléments supérieurs ou égaux à x est au moins N/2
 - le nombre d'éléments inférieurs ou égaux à x est au moins N/2
 On peut démontrer que tout ensemble possède une médiane (parfois plusieurs).
 Ecrire une fonction médiane qui prend en argument une liste L et qui retourne une médiane de L
4. Etant donné un vecteur $\vec{x} = (x_i)_{1 \leq i \leq n}$ la norme 1 de \vec{x} est définie comme $\|\vec{x}\|_1 = \sum_{i=1}^n |x_i|$ Ecrire une fonction norme1 qui prend en argument une liste L et qui retourne la norme 1 du vecteur représenté par L. Vous pouvez utiliser la fonction abs existant par défaut en Python et donnant la valeur absolue.
5. Etant donné un vecteur $\vec{x} = (x_i)_{1 \leq i \leq n}$ la norme 2 de \vec{x} est définie comme $\|\vec{x}\|_2 = \sqrt{\sum_{i=1}^n x_i^2}$ Ecrire une fonction norme2 qui prend en argument une liste L et qui retourne la norme 2 du vecteur représenté par L.

Exercice 4 (5 pts)

Complétez le programme conversion.py afin que le programme se lance de manière suivante dans le terminal :

```

Ce programme vous permet d'effectuer des conversions d'unités
Menu des Conversions:
1 - pouces vers cm
2 - cm vers pouces
2 - m vers cm
3 - cm vers m
3 - km vers miles
4 - miles vers km
4 - yard vers m
5 - m vers yard
5 - kg vers livres
6 - livres vers kg
Donnez une valeur entre 1 et 6 : 1
Conversion pouces -> cm. Donnez la valeur en pouces (ou 'q' pour quitter) : 10
Résultat de la conversion : 10.0 pouces = 25.4 cm
Conversion pouces -> cm. Donnez la valeur en pouces (ou 'q' pour quitter) : 20
Résultat de la conversion : 20.0 pouces = 50.8 cm
Conversion pouces -> cm. Donnez la valeur en pouces (ou 'q' pour quitter) : q

```

BONUS

Exercice 1

1. On considère la fonction f de deux variables x et y définie par :

$$\forall x, y, f(x, y) = \begin{cases} x^2 - x * y + 3 & \text{si } x > y \\ y^3 + x - 2 & \text{si } x < y \\ y + 3 & \text{sinon.} \end{cases}$$

Ecrivez une fonction qui prend en entrée deux réels x et y puis retourne la valeur de $f(x, y)$

2. En utilisant cette fonction, écrivez un programme qui demande à l'utilisateur de saisir deux nombres réels x et y , puis qui affiche le message suivant : "La valeur de la fonction f pour $x=5.3$ et $y=3.1$ est 14.66"
(ici 5.3 et 3.1 sont les deux valeurs saisies par l'utilisateur).
3. On considère à présent la suite $(u_n)_{n \in \mathbb{N}}$ définie par $u_0 = 0$ et, pour tout $n \geq 0$, $u_{n+1} = f(0, u_n)$
4. En utilisant la fonction définie à la question 1, écrivez un programme demandant à l'utilisateur de saisir un entier naturel n et affichant la valeur du terme U_n .

5. Modifiez le programme précédent pour qu'il affiche, étant donné un entier naturel n saisi par l'utilisateur, le plus grand des termes $u_0, u_1, u_2, \dots, u_n$

Exercice 2

La formule suivante, due à Leibniz en 1682, donne une méthode de calcul de π :

![[Pasted image 20240121163923.png]]

1. Ecrire une fonction `terme` qui prend en entrée un entier naturel n et retourne la valeur $v_n = \frac{(-1)^n}{2n+1}$
2. Ecrire une fonction `approx` qui prend en entrée un entier naturel n et qui retourne la valeur $4 \sum_{i=1}^n v_i$

`conversion.py` :

```
"""  
  
Collections  
  
- liste de conversions possibles  
  
- pouces -> cm (...)  
  
- cm -> m  
  
- km -> miles  
  
"""
```

```
# unité de départ, unité d'arrivée, facteur de conversion
```

```
CONVERSIONS = (                                #   choix   index   pair
(choix-1)//2
```

```
    ("pouces", "cm", 2.54),                      # 1       0       F
0
```

```
                                # 2       0 + R       T
0
```

```
    ("m", "cm", 100),                            # 3       1
1
```

```
                                # 4       1 + R
1
```

```
    ("km", "miles", 0.621371),                   # 5       2
2
```

```
                                # 6       2 + R
2
```

```
    ("yard", "m", 0.9144),
```

```
    ("kg", "livres", 2.20462),
```

```
)
```

```
def conversion_menu(conversions):
```

```
    """
```

Affiche le menu des conversions et renvoie le choix de l'utilisateur.

Parameters:

conversions (tuple): Liste des conversions possibles.

Returns:

int: Choix de l'utilisateur.

```
"""
```

```
pass
```

```
def demander_valeur_numerique_utilisateur(valeur_min,  
valeur_max):
```

```
    """
```

Demande à l'utilisateur de saisir une valeur numérique dans une plage spécifiée.

Parameters:

valeur_min (int): Valeur minimale autorisée.

`valeur_max (int): Valeur maximale autorisée.`

Returns:

`int: Valeur saisie par l'utilisateur.`

`"""`

`pass`

`def demander_et_afficher_conversion(unit1, unit2, facteur,
reverse=False):`

`"""`

Demande à l'utilisateur la valeur à convertir et affiche le
résultat de la conversion.

Parameters:

`unit1 (str): Unité de départ.`

`unit2 (str): Unité d'arrivée.`

`facteur (float): Facteur de conversion.`

`reverse (bool, optional): Indique si la conversion inverse
doit être effectuée. Par défaut, False.`

Returns:

bool: True si l'utilisateur souhaite quitter, False sinon.

```
"""
```

```
pass
```

```
# Menu principal
```

```
def main():
```

```
    print("Ce programme vous permet d'effectuer des  
conversions d'unités")
```

```
    # Menu : choix de la conversion
```

```
    # .....
```

```
    # Demander les valeurs à convertir à l'utilisateur
```

```
    # .....
```

```
if __name__ == "__main__":
```

```
    main()
```