

DEVOIR D'INFORMATIQUE N° TP 1 (1 HEURES)

Ce TP noté est constitué de plusieurs petits exercices. Vous avez le droit d'utiliser les documents de cours ainsi que vos documents personnels mais, bien évidemment, pas ceux de vos voisins. Vous rendrez vos réponses sur la feuille de réponses jointes en y écrivant les valeurs ou réponses aux questions posées, les fonctions ou instructions Python qui sont demandées etc... Vous utiliserez Pyzo et effectuerez les imports de toutes les fonctions du module `math` ainsi que le module `numpy` avec l'alias `np`. De plus, pour vérifier le caractère personnel de votre travail, vous enregistrerez tous les quarts d'heure vos fichiers Python dans le dossier **Mes devoirs** d'où ils seront ramassés de manière régulière.

Attention : Pour les applications numériques ou d'autres questions (dont la première), vous possédez chacun un identifiant personnel (M, K) couple de deux nombres donnés dans le tableau suivant :

Eleve	M	K	Eleve	M	K	Eleve	M	K
Aymon	21	13	Ferant	21	14	Lepinay	21	15
Besnard	21	16	Chevalier	21	17	Domanchin	21	18
Carre	22	13	Deparday	22	14	Jeanneret	22	15
Krifa	22	16	Lucas	22	17	Rousseau	22	18
Danner	23	13	Niang	23	14	Sylla	23	15
Boulard	23	16	Luciani	23	17	Dziedziczak	23	18

Préliminaires : Utilisation de vos identifiants

Nous allons utiliser vos identifiants (M, K) pour ce TP, afin de déterminer 3 listes sur lesquelles vous effectuerez les différents tests des fonctions programmées

1. Ecrire des séquences d'instructions permettant d'obtenir la liste **maliste** constituée des 100 entiers démarrant de M et séparés de K . Par exemple si on avait pris $M = 10$ et $K = 2$, on aurait eu la liste des 100 entiers pairs compris entre 10 et 208.
2. Ecrire une fonction numérique **f1** qui reçoit comme argument un nombre entier ou flottant x et qui retourne l'entier $\lfloor 100 \times (\ln(x) - \lfloor \ln(x) \rfloor) \rfloor$
3. En appliquant la fonction précédente **f1** à votre liste **maliste**, déterminer la liste **maliste2** des $f1(x)$ lorsque x décrit la liste **maliste**. Sur la feuille des réponses, vous noterez les 5 premiers termes et les 5 derniers termes de la liste **maliste2** dans l'ordre où ils sont obtenus (donc dans l'ordre de la liste **maliste**).
4. Ecrire une fonction numérique **f2** qui reçoit comme argument une liste d'entiers *Liste* et qui retourne la liste ordonnée des entiers compris entre 1 et 50 et qui ne sont pas dans la liste *Liste*
5. Votre troisième liste est la liste **maliste3** constituée du résultat de **f2** pris en **maliste2**. Sur la feuille des réponses, vous noterez la liste **maliste3** en entier.

Exercice 1 : Travail sur une longue chaîne de caractères

Vous devez enregistrer votre travail dans le dossier **Mes devoirs** pour avoir accès à un fichier nommé `texte.py` qui contient le début de l'oeuvre **Les Perses**.

A l'aide de la commande

```
>>> from texte import texteOr, texte
```

placée dans votre script, vous avez accès à deux chaînes de caractères **texteOr** et **texte** contenant les chaînes de caractères correspondant à ce texte : **texteOr** est le texte original alors que **texte** est la même chaîne de caractères mais où on a enlevé les accents (mais en gardant les lettres) et changer les majuscules en minuscules.

1. Créer la chaîne de caractère **chaine1** composée des caractères de **texte** se trouvant aux places des entiers figurant dans **maliste**. Faites de même pour créer **chaine2** et **chaine3** extraites de **texte** à l'aide des listes **maliste2** et **maliste3**. Sur la feuille de réponses, vous indiquerez les 10 premiers caractères ainsi que les 10 derniers de ces trois chaînes. Attention : les signes de ponctuations, le passage à la ligne, noté `\n`, et l'espace comptent pour un caractère.

2. Ecrire une fonction ou une séquence d'instructions, permettant de dénombrer le nombre de "b" dans la chaîne de caractères **texte**.
3. Ecrire une fonction ou une séquence d'instructions, permettant de dénombrer le nombre de voyelles minuscules dans la chaîne de caractères **texteOr**.

Exercice 2 : Suites de Syracuse pour les nombres entiers négatifs On part d'un nombre entier négatif n non nul s'il est pair, on le divise par 2 s'il est impair, on le multiplie par 3 et on ajoute 1. En répétant l'opération, on obtient une suite d'entiers positifs dont chacun ne dépend que de son prédécesseur. Par exemple, en partant de $n = -12$, les 10 premiers termes de la suite sont : -12, -6, -3, -8, -4, -2, -1, -2, -1, -2. Comme pour le cas positif, on ne sait pas si toute suite démarrante avec un entier n négatif reste bornée ou non. Par contre, on sait que toutes les suites ne finiront pas nécessairement sur le même cycle : en effet la suite démarrante à -12 se termine par le cycle : (-2, -1) alors que la suite démarrante à -17 se termine par le long cycle : (-17, -50, -25, -74, -37, -110, -55, -164, -8, -41, -122, -61, -182, -91, -272, -136, -68, -34, -17).

On écrit la fonction **syr** qui permet de calculer le terme suivant dans la suite.

```
>>> def syr(n):
>>>     if n % 2 == 0:
>>>         return(n//2)
>>>     else:
>>>         return (3*n + 1)
```

1. Ecrire une fonction **suitesyracuse50** prenant comme argument un entier relatif n et retournant la liste des 50 premiers termes de la suite de Syracuse débutant à n .
2. Modifier la fonction précédente pour obtenir une fonction **suitesyracuse** prenant comme argument un entier relatif n et retournant soit la liste des 50 premiers termes de la suite de Syracuse débutant à n si ces termes sont tous différents, soit la liste de tous les termes distincts 2 à 2 dans l'ordre d'obtention dans le cas contraire.
3. Appliquer la fonction **suitesyracuse** à chacun des opposés des nombre de la liste **maliste3**. Parmi ces suites, combien y-en a-t-il se terminant sur le cycle (-17, -50, -25, ...)? Sur le cycle (-2, -1)? Y-a-t-il d'autres terminaisons possibles?

DEVOIR D'INFORMATIQUE N° TP 1 (1 HEURES)

Ce TP noté est constitué de plusieurs petits exercices. Vous avez le droit d'utiliser les documents de cours ainsi que vos documents personnels mais, bien évidemment, pas ceux de vos voisins. Vous rendrez vos réponses sur la feuille de réponses jointes en y écrivant les valeurs ou réponses aux questions posées, les fonctions ou instructions Python qui sont demandées etc... Vous utiliserez Pyzo et effectuerez les imports de toutes les fonctions du module `math` ainsi que le module `numpy` avec l'alias `np`. De plus, pour vérifier le caractère personnel de votre travail, vous enregistrerez tous les quarts d'heure vos fichiers Python dans le dossier **Mes devoirs** d'où ils seront ramassés de manière régulière.

Attention : Pour les applications numériques ou d'autres questions (dont la première), vous possédez chacun un identifiant personnel (M, K) couple de deux nombres donnés dans le tableau suivant :

Eleve	M	K	Eleve	M	K	Eleve	M	K
Gheerbrant	21	13	Geslin	21	14	Leve N.	21	15
Bernard	21	16	Giovannini	21	17	Lorentz	21	18
Brasseur	22	13	Olympio	22	14	Tran	22	15
Level S.	22	16	Louveau	22	17	Potokar	22	18
Barriac	23	13	Le Ruz	23	14	Louis	23	15
Amiot	23	16	Bibrac	23	17	Goubert	23	18

Préliminaire : Utilisation de vos identifiants

Nous allons utiliser vos identifiants (M, K) pour ce TP, afin de déterminer 3 listes sur lesquelles vous effectuerez les différents tests des fonctions programmées

1. Ecrire des séquences d'instructions permettant d'obtenir la liste **maliste** constituée des 100 entiers démarrant de M et séparés de K . Par exemple si on avait pris M = 10 et K = 2, on aurait eu la liste des 100 entiers pairs compris entre 10 et 208 .
2. Ecrire une fonction numérique **f1** qui reçoit comme argument un nombre entier ou flottant x et qui retourne l'entier $\lfloor 100 \times (\ln(x) - \lfloor \ln(x) \rfloor) \rfloor$
3. En appliquant la fonction précédente **f1** à votre liste **maliste**, déterminer la liste **maliste2** des $f1(x)$ lorsque x décrit la liste **maliste**. Sur la feuille des réponses, vous noterez les 5 premiers termes et les 5 derniers termes de la liste **maliste2** dans l'ordre où ils sont obtenus (donc dans l'ordre de la liste **maliste**).
4. Ecrire une fonction numérique **f2** qui reçoit comme argument une liste d'entiers *Liste* et qui retourne la liste ordonnée des entiers compris entre 1 et 50 et qui ne sont pas dans la liste *Liste*
5. Votre troisième liste est la liste **maliste3** constituée du résultat de **f2** pris en **maliste2**. Sur la feuille des réponses, vous noterez la liste **maliste3** en entier.

Exercice 1 : travail sur une longue chaîne de caractères

Vous devez enregistrer votre travail dans le dossier **Mes devoirs** pour avoir accès à un fichier nommé `texte.py` qui contient le début de l'oeuvre **Les Perses**.

A l'aide de la commande

```
>>> from texte import texteOr, texte
```

placée dans votre script, vous avez accès à deux chaînes de caractères **texteOr** et **texte** contenant les chaînes de caractères correspondant à ce texte : **texteOr** est le texte original alors que **texte** est la même chaîne de caractères mais où on a enlevé les accents (mais en gardant les lettres) et changer les majuscules en minuscules.

1. Créer la chaîne de caractère **chaine1** composée des caractères de **texte** se trouvant aux places des entiers figurant dans **maliste**. Faites de même pour créer **chaine2** et **chaine3** extraites de **texte** à l'aide des listes **maliste2** et **maliste3**. Sur la feuille de réponses, vous indiquerez les 10 premiers caractères ainsi que les 10 derniers de ces trois chaînes. Attention : les signes de ponctuations, le passage à la ligne, noté `\n`, et l'espace comptent pour un caractère.

2. Ecrire une fonction ou une séquence d'instructions, permettant de dénombrer le nombre de "b" dans la chaîne de caractères **texte**.
3. Ecrire une fonction ou une séquence d'instructions, permettant de dénombrer le nombre de voyelles minuscules dans la chaîne de caractères **texteOr**.

Exercice 2 : Suites de Syracuse pour les nombres entiers négatifs

On part d'un nombre entier négatif n non nul s'il est pair, on le divise par 2 s'il est impair, on le multiplie par 3 et on ajoute 1. En répétant l'opération, on obtient une suite d'entiers positifs dont chacun ne dépend que de son prédécesseur. Par exemple, en partant de $n = -12$, les 10 premiers termes de la suite sont : -12, -6, -3, -8, -4, -2, -1, -2, -1, -2. Comme pour le cas positif, on ne sait pas si toute suite démarrante avec un entier n négatif reste bornée ou non. Par contre, on sait que toutes les suites ne finiront pas nécessairement sur le même cycle : en effet la suite démarrante à -12 se termine par le cycle : (-2, -1) alors que la suite démarrante à -17 se termine par le long cycle : (-17, -50, -25, -74, -37, -110, -55, -164, -8, -41, -122, -61, -182, -91, -272, -136, -68, -34, -17).

On écrit la fonction **syr** qui permet de calculer le terme suivant dans la suite.

```
>>> def syr(n):
>>>     if n % 2 == 0:
>>>         return(n//2)
>>>     else:
>>>         return (3*n + 1)
```

1. Ecrire une fonction **suitesyracuse50** prenant comme argument un entier relatif n et retournant la liste des 50 premiers termes de la suite de Syracuse débutant à n .
2. Modifier la fonction précédente pour obtenir une fonction **suitesyracuse** prenant comme argument un entier relatif n et retournant soit la liste des 50 premiers termes de la suite de Syracuse débutant à n si ces termes sont tous différents, soit la liste de tous les termes distincts 2 à 2 dans l'ordre d'obtention dans le cas contraire.
3. Appliquer la fonction **suitesyracuse** à chacun des opposés des nombre de la liste **maliste3**. Parmi ces suites, combien y-en a-t-il se terminant sur le cycle (-17, -50, -25, ...)? Sur le cycle (-2, -1)? Y-a-t-il d'autres terminaisons possibles?

FEUILLE DE RÉPONSES Nom : Prénom :

Préliminaire : Utilisation de vos identifiants

1. Séquences d'instructions permettant d'obtenir la liste **maliste**.
2. Script de la fonction numérique **f1**
3. Noter les 5 premiers termes et les 5 derniers termes de la liste **maliste2**
4. Script de la fonction numérique **f2**
5. Noter la liste **maliste3** en entier.

Exercice 1 : Travail sur une longue chaîne de caractères

1. Notez les 10 premiers caractères ainsi que les 10 derniers de **chaîne1**
Notez les 10 premiers caractères ainsi que les 10 derniers de **chaîne2**

Notez les 10 premiers caractères ainsi que les 10 derniers de **chaîne3**

2. Fonction ou séquence d'instructions comptant le nombre de "b" dans **texte**.

Résultat de ce dénombrement :

3. Fonction ou séquence d'instructions comptant le nombre de voyelles minuscules dans **texteOr**.
Résultat de ce dénombrement :

Exercice 2 : Suites de Syracuse pour les nombres entiers négatifs

1. Script de **suitesyracuse50**
2. Script de **suitesyracuse**
3. Nombre d'éléments de **maliste3** dont les opposés engendrent une suite se terminant sur le cycle (-7, -20, -10, -5, -14) :

Nombre d'éléments de **maliste3** dont les opposés engendrent une suite se terminant sur le cycle (-2, -1)

Nombre de situations distinctes pour la terminaisons possibles pour les éléments de **maliste3**

CORRECTION