Les fonctions et les procédures (programmation Python)
TP

Exercice 1

Primalité

1. Écrire une fonction `facteurs_premiers(n)` qui renvoie la liste de tous les facteurs premiers d'un entier $2 \leq n$.
2. Écrire une fonction `liste_premiers(n)` qui renvoie la liste L de tous les nombres premiers strictements inférieurs à n.

Pangramme

1. Écrire une fonction Python `pangramme` pour vérifier si une chaîne de caractères est un pangramme ou non.
   Note : Les pangrammes sont des mots ou des phrases contenant chaque lettre de l'alphabet au moins une fois.
   Par exemple : "Le zéphyr tourne les pages de mon vieux dico Robert : quel show, je kiffe !"

Exercice 2

Calcul d'une suite

On considère la suite u définie par $u_0 = 0$, et pour tout entier n : $u_{n+1} = 2u_n + 1$
Q.1 Écrire une fonction `suite(n)` qui calcule tous les éléments de la suite jusqu'à u_n et les stocke dans une liste L tel que $L[k] = u_k$.
Q.2 Écrire une fonction `Somme_suite(N)` qui calcule la somme de tous les éléments de la suite jusqu'à u_N : $\sum_{n=0}^{N} u_n$

Génération par compréhension

On considère la génération d'une liste avec un programme de la forme :

```
def f(u):
 return .....
Objet=[......]
L=[f(u) for u in Objet]
```

Compléter les .... dans le programme ci-dessus pour générer les listes suivantes :
Q.1 $L = [2, 4, 8, 10, 12, 14, 16, 20]$
Q.2 $L =$ ['mon chat','ton chat','son chat','leur chat','le chat','ce chat']
Q.3 $L = [0, 1, 0, 1, 0, 1, 0, 1]$

Exercice 3

Une année **bissextile** est une année comportant 366 jours au lieu de 365 jours pour une année non bissextile.

Une année est bissextile, si elle est divisible par 4 mais pas par 100, ou bien si elle est divisible par 400. Les années sont bissextiles tous les quatre ans.
Exemple d'années bissextiles : 2016, 2020, 2024, 2028...

Travail demandé :

Ecrire une fonction `Bissextile(A)` permettant de tester si une année A est bissextile ou non.

Exercice 4 :
Un nombre heureux est un nombre entier qui, lorsqu'on ajoute les carrés de chacun de ses chiffres, puis les carrés des chiffres de ce résultat et ainsi de suite jusqu'à l'obtention d'un nombre à un seul chiffre égal à 1 (un).

Exemple :

$N = 7$ est `heureux`, puisque :

- $7^2 = 49$
- $4^2 + 9^2 = 97$
- $9^2 + 7^2 = 130$
- $1^2 + 3^2 + 0^2 = 10$
- $1^2 + 0^2 = 1$

On est arrivé à un nombre d'un seul chiffre qui est égal à 1, donc $N = 7$ est heureux.
Ecrire une fonction `heureux(nb)` qui permet de déterminer si un nombre entier `nb` est heureux ou non.

Exercice 5 : La suite de `Robinson` est définie par :

- $U_0 = 0$
- U_n se construit en concaténant le nombre d'apparition de chacun des chiffres constituant le terme U_{n-1} suivi du chiffre lui-même, selon l'ordre décroissant des chiffres, pour tout $n > 0$.

Exemple :

Pour $n = 5$, $U_5 = 13123110$

En effet :

- $U_0 = 0$
- $U_1 = 10$ car il y a une apparition (1) du chiffre 0 dans U_0
- $U_2 = 1110$ car il y'a une apparition (1) du chiffre 1 et une apparition (1) du chiffre 0 dans U_1
- $U_3 = 3110$ car il y'a une apparition (3) du chiffre 1 et une apparition (1) du chiffre 0 dans U_2
- $U_4 = 132110$ car il y'a une apparition (1) du chiffre 3, deux apparition du chiffre 1 et une apparition (1) du chiffre 0 dans $U_[3$
- $U_5 = 13123110$ car il y'a une apparition (1) du chiffre 3, une apparition du chiffre 2, trois apparitions du chiffre 1 et une apparition (1) du chiffre 0 dans U_3

Travail demandé :

Ecrire une fonction `Robinson(N)` permettant de calculer le Nième terme de la suite de robinson