FACULTÉ DES SCIENCES DE TUNIS
DEUXIÈME ANNÉE DU CYCLE PRÉPARATOIRE INTÉGRÉ
FILIÈRES : MI
ANNÉE UNIVERSITAIRE 2024-2025

TP3 : Les fonctions en PYTHON

Exercice 1 Suite de Syracuse

La suite de Syracuse est définie par :
— $U_{n+1} = U_n$ si $U_n = 1$
— $U_{n+1} = U_n/2$ si U_n est pair
— $U_{n+1} = 3U_n + 1$ sinon

On dit que cette suite atteint toujours 1 au bout d'un certain nombre d'itérations.

1. Écrire une fonction PYTHON nommée *terme_suivant* qui à un entier a donné calcule et retourne le terme suivant 1, $a/2$ ou $3a + 1$ selon les cas.
2. Écrire une fonction PYTHON nommée *nbr_iter* permettant de calculer le nombre d'itérations minimum nécessaire pour aboutir à 1 à partir d'un entier a donné.
3. Écrire une fonction PYTHON nommée *nbr_iter_min* prenant en entrée une liste L d'entiers et permettant de calculer le nombre d'itérations minimum nécessaire pour que tous les éléments de L aboutissent à 1.
4. Écrire un script qui permet de saisir une liste d'entiers et d'afficher le nombre d'itérations minimum nécessaire pour que tous les éléments de L aboutissent à 1 en considérant la suite Syracuse.

Exercice 2 Algorithme d'Euclide

1. Écrire une fonction PYTHON intitulée *euclide_iter* qui calcule le pgcd de deux entiers a et b non nuls tels que a > b en utilisant l'algorithme d'Euclide, de façon itérative.
2. Donner la version récursive de la fonction précédente.

Exercice 3 Exponentiation rapide

1. Écrire une fonction PYTHON permettant de saisir un entier a tel que $a \geq 0$.

2. Écrire une fonction PYTHON intitulée puissance_naïve(x,n) qui permet de calculer et de retourner x^n (sans utiliser l'opérateur de puissance).

3. Donner la version récursive de la fonction puissance_naïve(x,n).

4. Écrire une fonction PYTHON permettant de calculer x^n ($x \geq 0$ et $n \geq 0$) selon la méthode de l'exponentiation rapide définie comme suit, sachant que lorsque $n = 0$, $x^n = 1$:
   — $x^n = x^{n/2} \times x^{n/2}$ si n est pair,
   — $x^n = x \times x^{(n-1)/2} \times x^{(n-1)/2}$ sinon.

5. Donner la version récursive de la méthode de l'exponentiation rapide.

Exercice 4 Méthode de Horner

Considérons un nombre N tel que $N = a_{n-1}a_{n-2}...a_1a_0$, où n désigne le nombre de chiffres de N, le nombre N est bien représenté dans la base b si et seulement si $\forall i \in [0, n-1]$, nous avons $0 \leq a_i \leq b-1$. Le nombre N représenté dans la base b est noté $N_{(b)}$. Le résultat de la conversion du nombre $N_{(b)}$ en base 10 est noté $N'_{(10)}$. Une méthode de conversion d'une base b à la base 10 consiste à appliquer la formule de Horner suivante :

$$N_{(b)} = a_0 + b(a_1 + b(... + b(a_{n-2} + ba_{n-1}))) = N'_{(10)}$$

1. Écrire une fonction PYTHON qui permet de saisir un entier N et une base b ($2 \leq b \leq 8$).
2. Écrire une fonction qui permet de vérifier si un entier N est bien représenté en base b.
3. Écrire une fonction qui permet de retourner la conversion d'un entier N en base 10 selon la méthode de Horner.