

Exercice n°01 :

A un entier n strictement positif on associe $n/2$ si n est pair et $3n+1$ si n est impair.

En réappliquant cette transformation à l'entier obtenu, on définit un algorithme dit de Syracuse.

Bien qu'à ce jour personne n'en ait trouvé de démonstration, il semble que pour tout entier strictement positif de départ on finisse toujours par arriver à 1.

Exemple : valeur de départ $n = 12$

6 3 10 5 16 8 4 2 1

On doit appliquer 9 fois la transformation.

Écrire un programme qui permet de saisir une valeur de départ proposée par l'utilisateur, d'afficher la liste des entiers obtenus jusqu'à 1, ainsi que le nombre de fois qu'il est nécessaire d'appliquer la transformation pour y arriver.

Exercice n°02 :

Tout nombre positif de deux chiffres n , tel que (chiffre des dizaines \neq chiffre des unités), possède une liste appelée liste vers 9.

Le principe est le suivant : on calcule la différence entre n et son symétrique, le résultat trouvé subit le même traitement, on répète ce processus jusqu'à obtenir une différence 9. L'ensemble constitué par le nombre initial et les résultats des différences est appelé liste vers 9.

Exemples :

Soit $n = 18 \rightarrow |18-81| = 63 \rightarrow |63-36| = 27 \rightarrow |27-72| = 45 \rightarrow |45-54| = 9 \rightarrow$ fin du traitement

La liste vers 9 est la suivante : 18 63 27 45 9

Soit $n = 74 \rightarrow |74-47| = 27 \rightarrow |27-72| = 45 \rightarrow |45-54| = 9 \rightarrow$ fin du traitement

La liste vers 9 est la suivante : 74 27 45 9

Écrire un programme qui permet de d'introduire un nombre n positif composé de deux chiffres obligatoirement différents, de générer et d'afficher sa liste vers 9.

Exercice n°03 :

Soit la suite de Kevin définie pour tout $n > 0$ par : K_n = concaténation de K_{n-2} et K_{n-1}

Exemple :

$K_0 = "a"$ et $K_1 = "c"$

$K_2 = "ac"$

$K_3 = "cac"$

$K_4 = "accac"$

$K_5 = "cacaccac"$

Écrire un programme qui permet de saisir K_0 et K_1 (deux lettres alphabétiques minuscules distincts) et d'afficher les n premiers termes de la suite de Kevin (n donnée).

Exercice n°04 :

Cette suite se base sur la chaîne binaire (0 et 1), le principe de cette suite est de remplacer toute occurrence du caractère "0" par la chaîne "01" et toute occurrence du caractère "1" par la chaîne "10".

$U_0 = "0"$

$U_1 = "01"$

$U_2 = "0110"$

$U_3 = "01101001"$

Écrire un programme qui permet d'afficher les n premiers éléments de cette suite.

Exercice n°05 :

$U_0 = 3$

$U_1 = 0$

$U_2 = 2$

$U_n = U_{n-2} + U_{n-3}$ pour $n \geq 3$

Cette suite vérifie la propriété suivante :

Pour tout entier n , si le reste de la division entière de U_n par $n = 0$, alors n est un nombre premier.

Écrire l'algorithme d'un module qui permet de vérifier si un entier n est premier en utilisant la propriété décrite précédemment.