

Exercice : Unicorn

# Enoncé

L'objectif de cet exercice est de résoudre un ensemble de problème mathématique.

Vous disposez pour cela d'une solution comprenant 2 projets :

* ITI.Unicorn.Tests, contient les tests unitaires
* ITI.Unicorn.Core, contient un ensemble d’implémentation par défaut des différents problèmes

Pour faire tourner les tests unitaires il vous suffit de configurer le projet ITI.Unicorn.Tests en tant que projet de démarrage puis d'exécuter la solution.

Comme vous pouvez le constater, pour le moment, tous les tests sont rouges :

**INSERT IMAGE TEST**

A vous de faire en sorte qu'ils passent en vert. Pour cela, vous avez le droit de faire ce que bon vous semble dans le projet ITI.Unicorn.Core.

Les tests unitaires sont là pour spécifier de façon détaillée les fonctionnalités attendues. Cependant le présent document va décrire brièvement ce qui est attendu pour les différents problèmes.

# Implémentation des différents problèmes

Certains tests ont un attribut Timeout défini, indiquant un temps maximal autorisé en Milliseconde. Les limites de temps définies sont généreuses. Si votre test est rouge pour cause de timeout, revoyez votre implémentation.

## Level 1

### Unicornflakes ( a revoir)

Le but de l’exercice est de transformer un tableau de Int en un tableau de String en ayant les multiples de 3, 5 et 7 remplacés par des chaînes de caractères définies selon les critères suivants :

* Les multiples de 3 : uni
* Les multiples de 5 : corn
* Les multiples de 7 : flakes
* Les multiples de 3 et 5 : unicorn
* Les multiples de 3 et 7 : uniflakes
* Les multiples de 5 et 7 : cornflakes
* Les multiples de 3 et 5 et 7 : unicornflakes

Ecrire une fonction qui prends en argument un tableau de Int et retourne un tableau de String ayant subi les transformations ci-dessus.

### No lemons, no melon ( a revoir)

On cherche à trouver le plus grand palindrome produit par des facteurs ayant le même nombre de chiffre.

Ecrire une fonction qui prends en argument le nombre de chiffre des facteurs et retourne le plus grand palindrome.

### Optimus Prime

En parcourant le début de la liste des nombres premiers : 2, 3, 5, 7, 11, et 13, on peut voir que le 5ème nombre premier est le 11.

Ecrire une fonction qui prends en argument un nombre n et retourne le nombre premier.

### Lot of numbers

Ecrire une fonction qui prends en argument une String de chiffres et un nombre n. Cette fonction retourne le plus grand produit de n chiffre consécutif dans la String.

### Lot of numbers Sequel

Ecrire une fonction qui prends en argument un tableau à 2 dimension. Cette fonction retourne le plus grand produit de 4 nombres consécutifs contenu dans le tableau, que ce soit vers le bas, le haut, la droite, la gauche ou diagonalement.

## Level 2

### Rabbits

Ecrire une fonction qui prends en argument un nombre n et retourne la somme de tous les nombres pairs de la suite de Fibonacci inférieur à n.

Rappel : *La valeur d'un INT est compris entre -2,147,483,648 et 2,147,483,647, attention à l'overflow.*

### The big one

On définit une suite itérative par :

* n -> n /2 si n est pair
* n -> 3n +1 si n est impair

On s’arrête une fois que le nombre 1 est atteint.

Si le premier terme est 5, on obtient alors une séquence de 6 nombres tel quel :

5 → 16 → 8 → 4 → 2 → 1

Selon les règles définies précédemment, écrire une fonction qui prends en argument un nombre n et retourne le nombre compris entre 1 et n, qui utilisé comme terme initial, génère la plus grande séquence.

### Khéops

En commençant au sommet de la pyramide et descendant par les nombres adjacents de la ligne inférieur, trouvez le chemin produisant le plus grand résultat en sommant les nombres rencontrés.

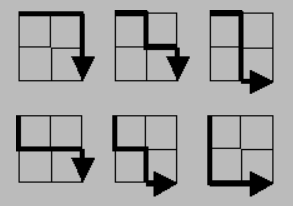


Ainsi, pour cette pyramide, le chemin le plus gros est 3 + 7 + 4 + 9 soit 23.

Ecrire une fonction qui prends en argument la pyramide (un tableau de tableau) et retourne la valeur du chemin produisant la plus grosse somme.

## Level 3

### Shortcut

Dans une grille de 2x2 où l’on peut se déplacer uniquement vers le bas et la droite, il existe 6 chemins différents pour arriver en bas à droite (en partant d’en haut à gauche) tel que :

Ecrire une fonction qui prends en argument un nombre a correspondant à la largeur du rectangle et b à sa hauteur. Cette fonction retourne le nombre de chemins possible pour traverser la grille.

Il existe au moins deux solutions différentes. Avec et sans boucle

à vous de jouer !