

## TP Algorithmique et programmation 1

### TP 5 - Tableaux

**Attention :** un tableau est caractérisé par l'adresse mémoire de sa première case. Du coup quand vous passez un tableau en paramètre vous passez cette adresse. Si vous modifiez un tableau dans une fonction, cela va écrire dans la mémoire et modifier le tableau du programme principal. Faites donc bien attention à ne pas modifier les tableaux passés en paramètre.

## Exercice 1 Traduction

Voici ci-dessous l'énoncé légèrement modifié d'un exercice de la feuille de TD4 sur les tableaux. Refaire cet exercice en Python.

Dans cet exercice on va manipuler un tableau *tab* d'entiers de dimension 100. Écrire un programme Python permettant de réaliser les tâches suivantes :

1. Initialiser le tableau *tab* puis saisir le nombre *nbcases* d'entiers que l'on veut mettre dans le tableau (si l'utilisateur donne une valeur inférieure ou égale à 0 ou strictement supérieure à 100 on affiche un message d'erreur et on redemande la valeur jusqu'à ce qu'elle soit correcte),
2. saisir les *nbcases* entiers et les stocker dans le tableau,
3. ajouter une fonction **est\_la** qui prend trois paramètres : *n* un nombre de cases, *t* un tableau et un entier *v*, et retourne un booléen qui est **Vrai** si et seulement si la valeur *v* est présente dans *t*, puis tester la fonction,
4. écrire une fonction **prem\_occ** qui prend les mêmes paramètres et retourne l'indice de la première occurrence de *v* dans *t* si elle existe, -1 sinon, puis la tester
5. ajouter une fonction **sequence** qui prend en paramètres *n*, *t* et *deb* et retourne l'indice auquel se termine la séquence croissante commençant à l'indice *deb* de *t* (*deb* est supposé  $< nbcases$ ).  
Exemple : Si *tab* = [3, 5, -6, -2, 6, 3, 8] alors **sequence**(7,*tab*,2) retourne 4 (en effet la séquence croissante est -6, -2, 6 et va de la case 2 à la case 4) , puis la tester,
6. ajouter une dernière fonction **diametre** qui calcule le diamètre du tableau (= la différence entre la plus grande et la plus petite valeur), puis la tester.  
Exemple : Si *tab* = [3, 5, -6, -2, 6, 3, 8] alors **diametre**(*tab*,7) retourne 14 (= 8 - (-6)).

## Exercice 2 Pour ceux qui ont fini : Crible d'Eratosthène

Le crible d'Eratosthène est une technique de calcul des nombres premiers inférieurs ou égaux à une borne *max* donnée. Le principe est de considérer tous les entiers de 2 à *max*. On parcourt ces entiers et on supprime tous les multiples de l'élément courant (sauf lui-même). Les éléments demeurant dans la liste en fin de traitement sont les nombres premiers inférieurs ou égaux à *max*.

Exemple : *max* = 20 donc les entiers de départ sont (2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20).

On rencontre 2 et l'ensemble devient (2, 3, 5, 7, 9, 11, 13, 15, 17, 19).

On rencontre 3 et l'ensemble devient (2, 3, 5, 7, 11, 13, 17, 19).

On rencontre 5, 7, 9, 11, 13, 17, 19 mais l'ensemble n'évolue plus. Écrire un programme Python qui :

1. initialise un tableau de 1001 booléens à la valeur **True**,
2. met les 2 premières cases (d'indice 0 et 1) à **False** ,
3. demande un entier *max* à l'utilisateur et réalise le crible d'Eratosthène sur ce tableau (parcourt tous les indices de 2 à *max* et pour chacun met à **False** toutes les cases dont l'indice est multiple du nombre courant (et donc pas premier)),
4. affiche tous les nombres premiers jusqu'à la borne *max*.