

---

## TP5 : Boucle while et recherche par dichotomie

---

Instructions introduites dans ce TP : `randrange`, `input`.

### Exercice TP5.1

On veut déterminer si une liste  $L$  contient une certaine valeur  $x$  sans nécessairement examiner tous ses éléments car on souhaite interrompre le parcours dès que l'élément est trouvé.

1. On donne les deux fonctions suivantes :

```
def mystere1(L,x):
    a=False
    for k in range(len(L)):
        if L[k]==x:
            a=True
    return(a)

def mystere2(L,x):
    s=False
    for k in range(len(L)):
        if L[k]==x:
            s=k
            break
    return(s)
```

Expliquer leur fonctionnement.

2. Réécrire les deux fonctions précédentes en utilisant une boucle `while`.

### Exercice TP5.2

1. Écrire une fonction `est_premier(p)` qui détermine si un entier  $p$  est premier en renvoyant `True` ou `False` et en utilisant une boucle `for`.
2. Écrire un programme `premiers()` qui renvoie la liste des 50 premiers nombres premiers avec une boucle `for` et une instruction `break`.
3. Refaire la question précédente avec une boucle `while`.

### Exercice TP5.3

On dit qu'une liste est *triée* si ses termes sont rangés dans l'ordre croissant. Par exemple, la liste `[1.3, 8.56, 10.2]` est triée alors que `[-12.86, 186.2, 151, 18.7]` ne l'est pas.

1. Écrire une fonction `Triee` utilisant une boucle `while` qui prend en entrée une liste  $L$  et qui renvoie `False` si la liste n'est pas triée, `True` sinon.
2. Écrire une fonction `IndiceObjetTriee` qui prend en entrée une liste triée  $L$  et un objet  $x$  et qui renvoie la position de l'objet dans la liste avec le moins de calculs possibles. On procédera par dichotomie.

### Exercice TP5.4

Dans cet exercice, `tableau` désigne un objet de type `ndarray`, de dimension 2.

1. Écrire une fonction qui prend en entrée un tableau et renvoie `True` si celui-ci comporte un terme nul, et `False` sinon.
2. Écrire une fonction qui prend en entrée deux tableaux de même taille et renvoie `True` si ceux-ci sont égaux, et `False` sinon.

**Exercice TP5.5**

On souhaite implémenter le jeu où l'on cherche à deviner un nombre entier compris entre 1 et 10. On utilise pour cela :

- la primitive `randrange()` de la librairie `random` : en effet, `randrange(n,p)` renvoie un nombre entier aléatoirement choisi entre  $n$  et  $p - 1$  ;
- la primitive `input()` qui permet à l'utilisateur d'entrer des données au clavier : la ligne de code

```
saisie = input("Proposez un entier")
```

affiche à l'écran le texte *Proposez un entier* et affecte ensuite à la variable *saisie* la valeur que l'utilisateur aura entrée au clavier (mais attention `input` convertira l'entier  $n$  entré en la chaîne de caractère ' $n$ ' ; pour reconvertir celle-ci en entier on utilisera la fonction `int`).

1. Écrire une procédure `Jeu` qui demande de saisir un nombre tant que le nombre mystère n'a pas été trouvé. Avant chaque nouvelle tentative, le programme doit afficher une indication du type *plus petit* ou *plus grand*.
2. Compléter votre programme pour qu'il demande au début du jeu le nombre maximal de tentatives autorisées.