TP1 RECHERCHE MINIMUM D'UNE LISTE

OBJECTIFS : Les objectifs de ce premier TP sont :

- de manipuler des listes simples avec python,
- d'écrire un algorithme qui permette de trouver le minimum d'une liste,
- d'intégrer cet algorithme dans une fonction,
- de déterminer la durée des différentes opérations réalisées lorsque la liste est de grande taille,
- d'écrire les valeurs d'une liste dans un fichier *.txt* .

DOCUMENT A RENDRE : Ce travail est évalué. Vous en rédigerez un compte-rendu au format *.pdf* pour le déposer en fin d'activité dans le répertoire *Devoir/TP1* du réseau avec le nom « tp1_nomfamille.pdf ». Ce compte-rendu contiendra :

- les réponses aux différentes questions posées,
- les copies d'écran des morceaux de codes écrits et celles des résultats des exécutions données dans le shell.

Pour commencer, ouvrir dans Pyzo un fichier nommé « recherche_mininmu.py » . Importer les bibliothèques random et time :

```
from random import *
import time
```

1. Rappels : création de listes

- Remplir une liste nommée « *petite_liste* » avec les valeurs suivantes : 174,407,287,984,933,527,627,449,503,722
- Créer une liste nommée « *grande_liste* » par compréhension qui contient 1000 valeurs de nombres entiers aléatoires compris entre 0 et 100 .
- On désire évaluer le temps de création de « *grande_liste* » grâce à la bibliothèque *time*.
  - Ouvrir la documentation https://docs.python.org/fr/3/library/time.html
  - Expliquer ce qu'est l'*epoch.*
  - Dans la console, exécuter : *import time* puis *time.gmtime(0)* . Quel est le moment que vous retourne cette commande ?
  - Exécuter à présent : *temps = time.time()* puis *time.gmtime(temps)* . Quel est le moment que vous retourne cette commande ?
  - Avant la création de « *grande_liste* » ranger la valeur du temps écoulé depuis l'epoch dans une variable nommée « *debut* ». Après cette création, faire de même dans une variable nommée « *fin* ».
  - Ecrivez ensuite le code permettant d'évaluer la durée de création de « *grande*_liste ».
- Compéter un tableau du type :

Taille de « *grande*_liste »	1000	10 000	100 000	1 000 000
Temps de création en s				

2. Evaluation du minimum : fonction et algorithme

- Ecrivez un code qui permet de trouver le minimum de la liste « *petite_liste* »
  - Par une structure *pour*
  - Par une structure *while*

Tester ces 2 codes qui doivent trouver la valeur 174 .

- Créer à présent une fonction nommée « *retourne_min(liste)* » dans laquelle vous inclurez le code précédent (structure *pour* seulement) afin que cette fonction retourne le minimum de la liste mise en argument.
  Par exemple : *minimum = calcul_min(petite_liste)* contiendra le nombre 174

- Employer cette fonction sur « *petite_liste* » et « *grande_liste* » et afficher à chaque fois un message du type « Le minimum est ….. et a été calculé en … s ». Pour cela, utiliser la méthode *.format* appliquées aux chaines de caractères dans la commande *print().*

- Créer à présent, une fonction nommée « *retourne_min_ind(liste)* » qui retourne la valeur du minimum et celle de l'indice correspondant. Commenter votre code pour indiquer le type d'objet retourné. Comme précédemment, utiliser cette fonction sur « *petite_liste* » et « *grande_liste* » et afficher les résultats dans la console.

3. Enregistrement des résultats dans un fichier

- Faites afficher les valeurs de « *grande_liste* » pour 1000 valeurs. Est-ce exploitable à l'écran ?
- Stocker à présent ces mêmes valeurs dans un fichier nommée « *grande_liste.txt* » qui sera configuré en écriture / lecture
- Ouvrir le fichier avec un éditeur de texte.
- Question subsidiaire ouvrir le fichier avec python et le faire afficher