# ÉVALUATION DE TP

## CI 2 : ALGORITHMIQUE ET PROGRAMMATION

**Consignes :**
- **tous les programmes réalisés seront enregistrés sous la forme Nom.Prenom.py ;**
- **ces programmes seront envoyés par mail lors des 5 dernières minutes de la séance ;**
- **les sujets seront restitués à la fin de la séance.**

**Objectifs des exercices 1 et 2 :**
- Alg – C1 : comprendre un algorithme et expliquer ce qu'il fait ;
- Alg – C2 : modifier un algorithme existant pour obtenir un résultat différent (ici modifier un algorithme pour obtenir un résultat similaire ;
- Alg – C4 : expliquer le fonctionnement d'un algorithme.

### Exercice 1 – Déchiffrer un programme Python – Sur feuille

On donne le programme suivant :

Pseudo Code

On a ;
- **données :** une liste à $N$ éléments numérotés de 0 à $N-1$, notée tab ;
- **variables :**
  - $i$ : compteur (nombre entier) ;
  - $tampon$ : variable de stockage (nombre réel) ;
- **fonctions :** $E(x)$ : fonction qui renvoie la partie entière de $x$.

**pour** *i allant de 0 à E(N/2)* **faire**
    tampon ← tab[i]
    tab[i] ← tab[N-1-i]
    tab[N-1-i] ← tampon
**fin**
afficher tab

On donne le programme suivant en Python :

python

```
1  tab = [17, 38, 10, 25, 72, 4, 98, 32, 11]
2  N = len (tab)
3  tampon = tab [N - 1]
4  for i in range (0, N - 1, 1) :
5      tab [N - 1 - i] = tab [N - 2 - i]
6  tab [0] = tampon
7  print (tab)
```

**Question 1**

*Expliquer ce que fait le programme précédent. Pour cela :*

- *décrire les instructions de chacune des lignes ;*
- *en utilisant un exemple simple, vous expliquerez comment évoluent chacune des variables ;*
- *vous donnerez l'objectif du programme.*

**Question 2**

*Sur feuille, proposer un programme Python réalisant la même tâche avec une boucle while à la place de la boucle for.*

## Exercice 2 – Déchiffrer un programme Python – Sur feuille

On donne le programme suivant en Python :

```python
tab = [4, 10, 11, 17, 25, 32, 38, 72, 98]
N = len(tab)
nb = 44
i = 0
while nb > tab[i] and i < N - 1 :
    i = i + 1
if i == N - 1 :
    tab.insert(N, nb)
else :
    tab.insert(i, nb)
print(tab)
```

**Question 1**

*Expliquer ce que fait le programme précédent. Pour cela :*

- *décrire les instructions de chacune des lignes ;*
- *en utilisant un exemple simple, vous expliquerez comment évoluent chacune des variables ;*
- *vous donnerez l'objectif du programme.*

## Exercice 3 – Suite de Syracuse – Sur PC

Objectifs :

- Alg – C3 : concevoir un algorithme répondant à un problème précisément posé ;
- Alg – C5 : écrire des instructions conditionnelles avec alternatives, éventuellement imbriquées ;
- Alg – C9 : choisir un type de données en fonction d'un problème à résoudre ;
- Alg – C10 : concevoir l'en-tête (ou la spécification) d'une fonction, puis la fonction elle-même ;
- Alg – C14 : documenter une fonction, un programme plus complexe.

Soit $u_0 = N$ avec $N$ un entier strictement positif. La suite de Syracuse est définie pour tout entier $n$ positif ou nul tel que $u_{n+1} = \frac{u_n}{2}$ si $u_n$ est pair, $3u_n + 1 sinon$.

**Question 1**

*Écrire une fonction Python appelée **syracuse**, prenant en paramètre un entier n et un entier positif $u_0$ retournant le $n^{ième}$ terme de la suite.*

**Question 2**

*Déterminer, en utilisant la fonction syracuse, les termes $U_{15}$, $U_{16}$, $U_{17}$, $U_{18}$ de la suite pour $u_0$ vallant 10, 15 et 20.*

**Question 3**

*Écrire une fonction Python appelée **syracuse_liste**, prenant en paramètre un entier $n$ et un entier positif $u_0$ retournant une liste contenant les $n$ premiers termes de la suite de syracuse.*

**Question 4**

*Vérifier que vos fonctions répondent aux objectifs Alg – C14.*

## Exercice 4 – Sur PC

**Question 1**

*Écrire deux fonctions Python, appelées produit_for et produit_while, prenant en paramètre un entier $n$ et retournant le produit des $n$ premiers entiers compris entre 1 (inclus) et $n$ (inclus). La fonction produit_for utilisera une boucle for tandis que la fonction produit_while utilisera une boucle while.*

**Question 2**

*Déterminer, en utilisant les fonctions produit_for et produit_while, el produit des entiers compris entre 1 et 10, entre 1 et 67 et entre 1 et 128.*

## Exercice 5 - Recherche d'un mot dans une chaîne de caractère – Sur PC

Objectifs :
– Alg – C11 : traduire un algorithme dans un langage de programmation.

Le but de la fonction suivante est de savoir combien de fois un mot apparaît dans une chaîne :

Pseudo Code

```
Données : texte (String), mot(String)
Début Fonction
    Recherche (mot,texte) :
    nb_mot ← 0
    pour i de 0 à longueur(texte) faire
        Si texte[i]=mot[0] Alors
            j←0
            Tant que j≠longueur(mot) Faire
                Si (i+j)≥ longueur(texte) Alors
                    retourner nb_mot
                Fin
                Sinon si texte[i+j] !=mot[j] Alors
                    break
                Fin
                j← j+1
            Fin
            Si j=longueur(mot) Alors
                nb_mot ← nb_mot+1
            Fin
        Fin
    fin
    retourner nb_mot
Fin
```

**Question 1**

*Retranscrire l'algorithme dans Python.*

**Question 2**

*Tester son bon fonctionnement.*