



EXAMEN DE FIN SEMESTRE 1

Matière : INFORMATIQUE

Classes : 1^{ère} Année BG Durée : 2 h

EXERCICE 1(3.5pts)

On appelle nombre d'Armstrong un nombre N égal à la somme des cubes de ses chiffres.

Exemple : $N = 153 = 1^3 + 5^3 + 3^3$

1) Ecrire une procédure **Test** qui reçoit un entier N composé de trois chiffres et qui renvoie dans une variable booléenne **Etat**, le résultat du calcul :

Etat $\left\{ \begin{array}{l} \text{Vraie si } N \text{ est un nombre d'Armstrong} \\ \text{Faux sinon} \end{array} \right.$

2) Ecrire l'algorithme principal permettant de calculer et d'afficher tous les nombres d'Armstrong compris entre 0 et 999 en appelant la procédure **Test**.

EXERCICE 2(4.5pts)

On se propose les 3 programmes suivants :

Total $\leftarrow 0$ Pour i de 1 à $(n-10)$ faire Pour j de $(i-10)$ à $(i+10)$ faire Total \leftarrow total+1 Fin pour Fin pour Ecrire(Total)	s=1 i=2 tanque $i \leq n$ faire s $\leftarrow s*i$ i $\leftarrow i+1$ Ecrire(s)	e $\leftarrow 1$ Pour i de 1 à n faire p $\leftarrow 1$ Pour j de 1 à i faire p $\leftarrow p*x/j$ Finpour e=e+p Finpour Ecrire(e)
--	--	--

Calculer la fonction Coût, $T(n)$ et en déduire la complexité $O(T)$ dans le pire des cas de chaque programme.

EXERCICE 3(3pts)

Une chaîne d'ADN est une chaîne de caractères qui contient que les lettres "a", "t", "c" et "g".

Ecrire un programme Python qui permet de :

1-Créer une liste vide « **adns** » et y ajouter les ADN ; "atcgta", "cct" et "agc" ; Afficher cette liste.

2- Créer et afficher un dictionnaire **d**, correspondant aux nombres d'apparitions de la lettre "c" dans chaque ADN de la liste **adns**.

Exemple si **adns**=["cta", "ccgt"] alors **d**={'cta': 1, 'accc': 3}

EXERCICE 4(9pts):

Suite de Syracuse : on rappelle la définition de la suite de Syracuse : U_0 est un entier positif et

$$\forall n \in \mathbb{N}, U_{n+1} \begin{cases} = U_n/2 & \text{si } U_n \text{ est pair} \\ = 3U_n+1 & \text{si } U_n \text{ est impair} \end{cases}$$

On donne $U_0=5$

Ecrire un programme python qui permet de :

1-Saisir un entier **n** tel que $1 \leq n \leq 10$.

2-Construit la liste **L**= [U_0, U_1, \dots, U_n] contenant les termes de la suite.

3-Calculer et afficher le minimum et le maximum des éléments de L.

4-Pour $n=5$ le programme affiche la liste suivante : **L**=[5, 16, 8, 4, 2, 1]

Ecrire les instructions python qui permettent d'effectuer les actions suivantes:

a- ajoutez l'élément 12 à la liste.

b- renverser la liste.

c- afficher l'indice de l'élément 16.

d- enlevez l'élément 8 de la liste.

e- afficher la sous-liste du 2^e au 3^e élément.

f- afficher la sous-liste du début au 2^e élément.

g- afficher la sous-liste du 3^e élément à la fin de la liste.

h- afficher la sous-liste complète de la liste.

i- afficher le dernier élément en utilisant un indexage négatif.