LYCÉE RÉGIONAL d'Enseignement Général et Technologique Arthur VAROQUAUX	python™ **ISN.** **TD 03**	**Term S**

Vous soignerez l'affichage pour l'ensemble des exercices

Exercice 1

Écrire un programme qui affiche les 20 premiers termes de la table de multiplication par 7, en signalant au passage (à l'aide d'un astérisque) ceux qui sont des multiples de 3.

Exercice 2

Ecrire un programme qui affiche une table de conversion de sommes d'argent exprimées en euros, en dollars américains. La progression des sommes de la table sera « géométrique » (1 euro , 2 euros, 4 euros, 8 euros etc…) Vous stopperez l'affichage pour les sommes d'argent supérieures à 500 euros.

Exercice 3

Écrire un programme qui convertisse un nombre entier de secondes fourni au départ en un nombre d'heures, de minutes et de secondes.
On utilisera les opérateurs // (division entière) et % (modulo).

Exercice 4

Écrire un programme qui affiche les 20 premiers termes de la suite numérique définie pout tout $n \in \mathbb{N}$ par : $\begin{cases} u_0 = a \\ u_{n+1} = 0{,}6u_n + 5 \end{cases}$
Où a est un nombre que l'utilisateur saisira. On affichera les termes sous la forme $(n\ ,\ u_n)$.

Exercice 5

On représente un fragment d'ADN par le chaîne de caractère : *adn = "agccgtaggctatttcgacgcaag"*

1. Ecrire un 1er programme qui :
   - affiche la chaîne d'ADN ?
   - affiche la longueur de cette chaîne d'ADN ?
   - affiche la première base de la chaîne d'ADN ?
   - ajoute le fragment "tga" au début de l'ADN ?
   - ajoute le fragment "ccc" à la fin de l'ADN ?
   - affiche la longueur de la nouvelle chaîne obtenue ?
2. Ecrire un 2e programme qui affiche toutes les bases une par une sur des lignes différentes.
3. Ecrire un 3e programme qui compte le nombre de fois qu'apparaît la base *a*.

Exercice 6

Ecrire un programme qui vérifie si la lettre ***e*** figure dans une chaine de caractères saisie par l'utilisateur.

Exercice 7

Ecrire un programme intercale un astérisque entre chaque caractère d'une chaîne.

Exercice 8 *(plus difficile)*

1. Ecrire un programme qui inverse une chaine de caractères (*abcd* → *dcba*)
2. Ecrire un programme qui vérifie si une chaîne de caractères est un palindrome. (ex : stats ou rotor)

LES CHAINES DE CARACTERES (Strings)

Une chaîne de caractères (*string*) représente du texte.

Elles s'écrivent entre guillemet : *nom = "Mr X"*

Demander à l'utilisateur d'entrer une chaîne :

*nom = **raw_input**("Entrez votre nom : ")*

Opérations sur les chaînes de caractères :

Concaténation de chaînes	*nom = "Jean"+"-Dupont"*	
Affichage d'une chaîne	*print nom*	……

c = "ceci est une chaine de caracteres"

Longueur d'une chaîne	*print **len**(c)*	……
Affichage d'un caractère	*print c[0]*	……
	print c[6]	……
	print c[-1]	……
Affichage d'une sous-chaîne	*print c[1 :4]*	……
	print c[2 :]	……
	print c[:2]	……

LES CHAINES DE CARACTERES (Strings)

Une chaîne de caractères (*string*) représente du texte.

Elles s'écrivent entre guillemet : *nom = "Mr X"*

Demander à l'utilisateur d'entrer une chaîne :

*nom = **raw_input**("Entrez votre nom : ")*

Opérations sur les chaînes de caractères :

Concaténation de chaînes	*nom = "Jean"+"-Dupont"*	
Affichage d'une chaîne	*print nom*	……

c = "ceci est une chaine de caracteres"

Longueur d'une chaîne	*print **len**(c)*	……
Affichage d'un caractère	*print c[0]*	……
	print c[6]	……
	print c[-1]	……
Affichage d'une sous-chaîne	*print c[1 :4]*	……
	print c[2 :]	……
	print c[:2]	……