

Fiche de TP/TD 1

(MSP2)

Informatique 3 : Programmation en Python

Conditions, boucles, chaînes

Exercice 1

(Somme cumulative)

Écrire un programme qui demande à l'utilisateur de saisir un entier $n > 0$, puis calcule et affiche la somme des entiers de 1 à n .

Variante : faire aussi la somme des carrés ou la somme des entiers pairs uniquement.

Exercice 2

Demander l'année de naissance, afficher l'âge (l'année courante est dans une variable).

Exercice 3

(Nombres pairs jusqu'à n)

Écrire un programme qui demande à l'utilisateur de saisir un entier $n > 0$, puis calcule et affiche sur une seule ligne tous les nombres pairs entre 0 et n (inclus), séparés par un espace.

Exercice 4

(Décompte inversé)

Écrire un programme qui demande à l'utilisateur de saisir un entier $n > 0$, puis calcule et affiche les nombres de n jusqu'à 1 (inclus) sur une ligne, séparés par un espace.

Exercice 5

(Budget & dépenses)

Écrire un programme qui demande à l'utilisateur de saisir un budget initial (entier). Puis lire des dépenses successives (entiers) tant que le budget restant est > 0 . Quand une dépense dépasse le budget restant, afficher le montant supplémentaire nécessaire pour la couvrir, puis arrêter.

Exercice 6

(Catalogue de températures)

Demander à l'utilisateur une température (float).

- Si > 20 : afficher « sortie en tenue d'été »
- Si < 7 : afficher « manteau nécessaire »
- Sinon : afficher « temps normal »

Variante : répéter la saisie jusqu'à ce que l'utilisateur tape « fin ».

Exercice 7

(Test d'âge & tarif)

Saisir l'âge d'une personne (entier).

Catégories : $\text{âge} \leq 11 \rightarrow \text{"Tarif enfant"}; 12 \leq \text{âge} \leq 27 \rightarrow \text{"Tarif jeune"}; \text{âge} \geq 60 \rightarrow \text{"Tarif senior"};$
sinon "plein tarif".

Variante : réaliser cette classification en utilisant le moins possible de branches imbriquées.

Exercice 8

(Division euclidienne par soustraction)

Écrire un programme qui demande à l'utilisateur de saisir deux entiers positifs a, b (avec $a > b$).

Calculer le quotient q et le reste r de la division de a par b en n'utilisant que des soustractions répétées (**while**).

Afficher q et r .

Exercice 9

(Logarithme entier en base 2)

Écrire un programme qui demande à l'utilisateur de saisir un entier $n \geq 1$, puis calcule et affiche le plus petit entier k tel que $2^k \geq n$, par une boucle multipliant par 2.

Variante : calculer aussi la plus grande puissance de $2 \leq n$.

Exercice 10

(Parité & conditions combinées)

Demander à l'utilisateur un entier n .

- Si n est pair *et* > 50 : afficher « pair et grand »
- Si n est pair *ou* multiple de 3 : afficher « pair ou multiple de 3 »
- Si $n \in [-13, 25]$: afficher « dans l'intervalle », sinon « hors intervalle ».

Travaillez les combinaisons de 'and' / 'or' / comparaisons.

Exercice 11

(Comptage et suppression d'un caractère)

Demander à l'utilisateur de saisir une chaîne de caractères s .

- Compter le nombre d'occurrences du caractère 'e' dans s .
- Construire une nouvelle chaîne sans le(s) caractère(s) 'e'.

Variante : faire le même traitement pour toutes les voyelles (a, e, i, o, u, y).

Exercice 12 (Moyenne)

Saisir des nombres jusqu'à **Entrée** vide, afficher la liste, puis la moyenne.

Exercice 13 (Chiffrement de César)

Écrire un programme qui intègre les options suivantes en considérant un décalage de 23 :

1. chiffrement d'une phrase saisie ;
2. déchiffrement d'une phrase chiffrée ;
3. version avec `str.maketrans` et `str.translate`.

Aide : `ord()`, `chr()`.

Exercice 14

(Concaténation maximale sans doublons)

Demander à l'utilisateur de saisir deux chaînes de caractères s_1 et s_2 . Construire une chaîne résultante maximale qui contient tous les caractères de s_1 puis ceux de s_2 *sans répétition* de caractères déjà présents. Ex : $s_1 = \text{"abracadabra"}$, $s_2 = \text{"barbecue"}$ → résultat possible : “abrcdeu” (ou toute variation respectant l'ordre minimal).

Exercice 15

(Compteur d'apparitions imbriquées)

Faire saisir une chaîne s . Pour chaque caractère c de l'alphabet (ou chaque lettre minuscule rencontrée dans s), compter et afficher combien de fois la paire “ cc ” (deux fois le même caractère consécutif) apparaît dans s . Ex : $s = \text{"balloon"}$ → pour ‘l’ on compte la paire “ll”, pour ‘o’ on compte “oo”. Variante : généraliser à paires différentes, ou comptage de triplets consécutifs.

Exercice 16

(Alternance de caractères)

Faire saisir une chaîne s . Déterminer si on peut “réarranger” les caractères de s (en permutant) pour obtenir une chaîne où aucun caractère n'apparaît deux fois de suite (i.e. pas de “aa”, “bb”, etc.). Si c'est possible : renvoyer “Possible”, sinon “Impossible”. Indice : calculer les fréquences, vérifier que la fréquence maximale n'est pas trop grande par rapport aux autres.

Exercice 17

(Somme croissante jusqu'à condition)

Saisir un entier positif N . Construire une boucle ‘while’ qui accumule des entiers consécutifs $1 + 2 + 3 + \dots$ jusqu'à ce que la somme dépasse N . Afficher à la sortie : le dernier terme ajouté k , la somme totale, et combien on a dépassé N . Ex : si $N = 50$, on calcule $1 + 2 + 3 + \dots + 10 = 55 (>50)$ → afficher $k = 10$, somme = 55, dépassement = 5.

Exercice 18

(Intersection de plages entières)

Faire saisir 3 entiers a, b, c (positifs). Considérer les trois intervalles entiers $[1, a]$, $[1, b]$ et $[1, c]$. Par une boucle(s) et conditions, déterminer et afficher :

- l'intersection entre $[1, a]$ et $[1, b]$,
- l'intersection entre $[1, a]$ et $[1, c]$,
- l'intersection entre $[1, b]$ et $[1, c]$,
- l'intersection commune des trois.

L'affichage se fait sous forme de listes d'entiers (ex : “[1,2,3]”).