



Université Hassan II de Casablanca
Faculté des Sciences & Techniques Mohammedia

Module TNIM
Prof. Abdellah ADIB

Lundi 09 Octobre 2023
Travaux pratiques

Prends le temps de réfléchir, mais lorsque c'est le moment de passer à l'action, cesse de penser et vas-y.

Andrew Jackson

Série 1

Exercice 1 (Les bases de Python)

Écrire un programme en Python qui :

- (1). Demande à l'utilisateur de saisir un nombre entier \mathcal{L} et lui informe si ce nombre est carré parfait ou pas.
- (2). Permet de calculer le \cos d'un angle par son approximation par série de Taylor :

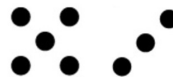
$$\cos(\alpha) = \sum_{n=0}^{+\infty} \frac{(-1)^n}{(2n)!} \alpha^{2n}$$

Évaluer la valeur de n permettant d'approcher la valeur de $\cos(\alpha)$ et qui affiche la vraie valeur, celle estimée et l'erreur commise.

- (3). Demande à l'utilisateur de saisir un nombre entier \mathcal{D} et lui affiche tous ses diviseurs.
- (4). On lance deux dés à six face et on souhaite afficher le résultat obtenu, les deux faces supérieurs, par la Figure 1 (A) et qu'on souhaite afficher sous forme de points égalant le chiffre obtenu sous la forme donnée par la Figure 1 (B).



A



B

Figure 1 –

- (5). Permet d'afficher tous les entiers positifs de trois chiffres de la forme cdu tel que, pour chaque entier, la somme de ses chiffres ($c+d+u$) est un diviseur du produit de ses chiffres ($c*d*u$).

Exemple : L'entier 514 vérifie cette propriété, en effet, $(5+1+4) = 10$ est un diviseur de $(5 * 1 * 4) = 20$.

- (6). Un nombre premier \mathcal{N} est dit circulaire s'il vérifie la propriété suivante : chacune des rotations de ses chiffres d'un élément vers la droite, forme à son tour un nombre premier.

Exemple : $\mathcal{N} = 719$ est un nombre premier circulaire car 719, 971 et 197 sont des nombres premiers.

Exercice 2 (Listes & Dictionnaires)

- (1). Étant donnée la liste des notes des élèves :

$$notes = [12, 04, 14, 11, 18, 13, 07, 10, 05, 09, 15, 08, 14, 16]$$

Écrivez un programme Python qui permet d'extraire de cette liste les notes au dessus de la moyenne et de les mettre dans une nouvelle liste.

- (2). Écrivez un programme Python qui permet de regrouper dans une liste les mots communs à deux chaînes $Ch1$ et $Ch2$.

- (3). Considérons le dictionnaire suivant :

First_dict = { "Appareil" : "Laptop" , "Marque" : "IBM" , "Carte mère" : "MSI Z490" , "Carte Graphique" : "GeForce RTX 3070" , "RAM" : "24G" , "Processeur" : "Intel core i7-G12" , "SSD" : "1 To" }

Écrire un programme qui permet de :

- (i) Corriger l'erreur "RAM" : "32G".
- (ii) Afficher la liste des clés, la liste des valeurs et la liste des paires de clés et valeurs.
- (iii) Inverser les paires "Processeur" : "Intel core i7-G12" et "Carte Graphique" : "GeForce RTX 3070".
- (iv) Ajouter la pair clé-valeur : "Système d'exploitation" : "Windows 11".
- (v) Transformer le dictionnaire en une liste de liste.

Exercice 3 (Fonctions & Fichiers)

- (1). Écrire une fonction qui retourne la colonne avec la plus grande somme d'une matrice.
- (2). Écrire une fonction Python permettant à partir d'une liste donnée de créer un fichier texte dont les lignes sont les éléments de cette liste.
- (3). Écrire une fonction en Python qui prend en paramètre un tuple formé de deux chaînes (S_1, S_2) et qui renvoie la liste des caractères de la chaîne S_1 qui ne se trouve pas dans la chaîne S_2 .
- (4). Un nombre est dit parfait lorsque il est égal à la somme de ses diviseurs propres. Un diviseur propre est un diviseur autre que le nombre lui-même. Le premier nombre parfait est 6. En effet 1, 2 et 3 sont les diviseurs propres de 6 et $6 = 1 + 2 + 3$. Écrire une fonction :

- (a) booléenne $check(n)$ qui teste si un entier n est parfait ou pas.
- (b) $ListPerfect(m)$ qui renvoie la liste des nombres parfaits strictement inférieurs à m .

- (c) Verify(n) qui, si n est un nombre parfait, retourne une liste de ses diviseurs propres vérifiant son égalité à leur somme.
- (d) Listverify(n) qui retourne une liste de liste des diviseurs propres de chaque éléments propres.
- (e) (Perfect) définissant quatre fonctions : check, ListPerfect, Verify et Listverify et tester toutes ses méthodes.
- (5). Supposons qu'il y ait 11 élèves et 10 questions, et que les réponses soient stockées dans une liste bidimensionnelle. Chaque ligne enregistre les réponses d'un élève aux questions, comme indiqué dans l'illustration suivante :

	0	1	2	3	4	5	6	7	8	9
<i>Etudiant 0</i>	E	D	D	A	C	B	E	E	A	D
<i>Etudiant 1</i>	A	E	C	C	E	A	B	D	D	A
<i>Etudiant 2</i>	B	C	B	B	A	E	E	D	E	D
<i>Etudiant 3</i>	C	A	E	A	A	D	B	B	E	B
<i>Etudiant 4</i>	B	C	C	A	E	B	A	B	C	E
<i>Etudiant 5</i>	D	E	C	A	E	C	C	B	A	E
<i>Etudiant 6</i>	E	A	B	B	A	C	E	A	A	B
<i>Etudiant 7</i>	E	C	C	B	B	B	D	D	A	C
<i>Etudiant 8</i>	C	E	E	A	B	C	D	D	E	A
<i>Etudiant 9</i>	D	A	C	C	B	A	B	A	E	E
<i>Etudiant 10</i>	A	C	C	B	B	E	A	E	A	B

La clé est stockée dans une liste unidimensionnelle :

0	1	2	3	4	5	6	7	8	9
A	D	D	C	C	A	E	B	C	E

- (a) Écrire une fonction qui compare les réponses de chaque élève avec la clé, compte le nombre de réponses correctes et les affiche.
- (b) Écrire une fonction qui affiche les élèves dans l'ordre croissant du nombre de bonnes réponses.
- (6). On souhaite gérer l'utilisation des locaux et créneaux pour une formation donnée, alors écrire une fonction qui :
- (a) Crée un dictionnaire contenant les numéros de cours et les numéros des salles où se déroulent les cours. Le dictionnaire doit contenir les paires clé-valeur suivantes :
- $$\{\text{"Num_Cours"} : \text{valeur}\} \text{ et } \{\text{"Num_Sal"} : \text{valeur}\}.$$
- (b) Crée un dictionnaire contenant les numéros de cours et les noms des instructeurs qui enseignent chaque cours. Le dictionnaire doit contenir les paires clé-valeur suivantes :
- $$\{\text{"Num_Cours"} : \text{valeur}\} \text{ et } \{\text{"Nom_Instruct"} : \text{chaîne de caractères}\}.$$
- (c) Crée un dictionnaire contenant les numéros de cours et la date et l'heure de réunion de chaque cours. Le dictionnaire doit contenir les paires clé-valeur suivantes :
- $$\{\text{"Num_Cours"} : \text{valeur}\} \text{ et } \{\text{"Dat_Heur_Reun"} : \text{Date}\}.$$
- (d) Écrire un programme qui permet à l'utilisateur d'entrer un numéro de cours, puis lui affiche le numéro de la salle de cours, l'instructeur et le créneau de la réunion.

Exercice 4 (Application)

Un mot de passe est une chaîne de caractères composée d'un ensemble de lettres accentuées et ne renferme aucun espace. La force d'un mot de passe varie en fonction de l'évolution d'un score \mathcal{S} selon les règles suivantes :

- ✓ Si $\mathcal{S} < 20 \implies$ mot du passe '**Très faible**'.
- ✓ Si $20 \leq \mathcal{S} < 40 \implies$ mot du passe '**Faible**'.
- ✓ Si $40 \leq \mathcal{S} < 80 \implies$ mot du passe '**Fort**'.
- ✓ Si $\mathcal{S} \geq 80 \implies$ mot du passe '**Très fort**'.

Le score \mathcal{S} est évalué par cumule des bonus b_i et soustraction des pénalités p_i selon :

$$\mathcal{S} = \sum_{i=1}^4 b_i - \sum_{i=1}^2 p_i$$

où

- b_1 = Nombre total de caractères * 4,
- b_2 = (Nombre total de caractères – nombre de lettres majuscules) * 2,
- b_3 = (Nombre total de caractères – nombre de lettres minuscules) * 3,
- b_4 = Nombre de caractères non alphabétiques * 5,

et

- p_1 = La longueur de la plus longue séquence de lettres minuscules * 2,
- p_2 = La longueur de la plus longue séquence de lettres majuscules * 3,

Travail demandé :

Écrivez une fonction qui retourne

- (1). le nombre de caractères majuscules.
- (2). le nombre de caractères non alphabétiques.
- (3). la longueur de la plus longue séquence de lettres majuscules.
- (4). la longueur de la plus longue séquence de lettres minuscules.
- (5). qui affiche le score d'un mot de passe

Écrire un programme, qui permet de mesurer la force d'un mot de passe donné.

