## Institut Universitaire des Sciences (IUS)

## Faculté des sciences et de technologies

**TD2 - Math-Info**

### **Préparé par :** Pierre Durell Naguiby BYRON.

## **Niveau:** L3 - Sciences Informatiques

Demander à l’utilisateur d’entrer les ventes d’un magasin pour 7 jours. Calculer le total et la moyenne des ventes. Afficher un graphe en ligne montrant l’évolution des ventes au fil de la semaine.

import matplotlib.pyplot as plt  
  
ventes\_journalieres = []  
somme = 0  
  
print("Veuillez entrer les ventes de votre magasin pour les 7 derniers jours.")  
  
for i in range(7):  
 while True:  
 try:  
 jour\_vente = float(input(f"Entrer la vente du jour {i+1} : "))  
 if jour\_vente < 0:  
 print("Les ventes ne peuvent pas être négatives. Veuillez réessayer.")  
 else:  
 ventes\_journalieres.append(jour\_vente)  
 somme += jour\_vente  
 break  
 except ValueError:  
 print("Entrée invalide. Veuillez entrer un nombre pour les ventes.")  
  
print(f"\nLe total des ventes est : {somme:.2f} Gourdes")  
moyenne = somme / 7  
print(f"La moyenne des ventes est : {moyenne:.2f} Gourdes")  
  
  
## Évolution des ventes au fil de la semaine  
  
jours = [f"Jour {i+1}" for i in range(len(ventes\_journalieres))]  
  
plt.figure(figsize=(10, 6))  
  
plt.plot(jours, ventes\_journalieres, marker='o', linestyle='-', color='b', label='Ventes journalières')  
  
plt.axhline(y=moyenne, color='g', linestyle='--', label=f'Moyenne ({moyenne:.2f} Gourdes)')  
  
plt.xlabel("Jour de la semaine")  
plt.ylabel("Ventes (Gourdes)")  
plt.title("Évolution des ventes au fil de la semaine")  
plt.legend()  
plt.grid(True)  
plt.ylim(bottom=0)  
plt.xticks(rotation=45)  
plt.tight\_layout()  
plt.show()

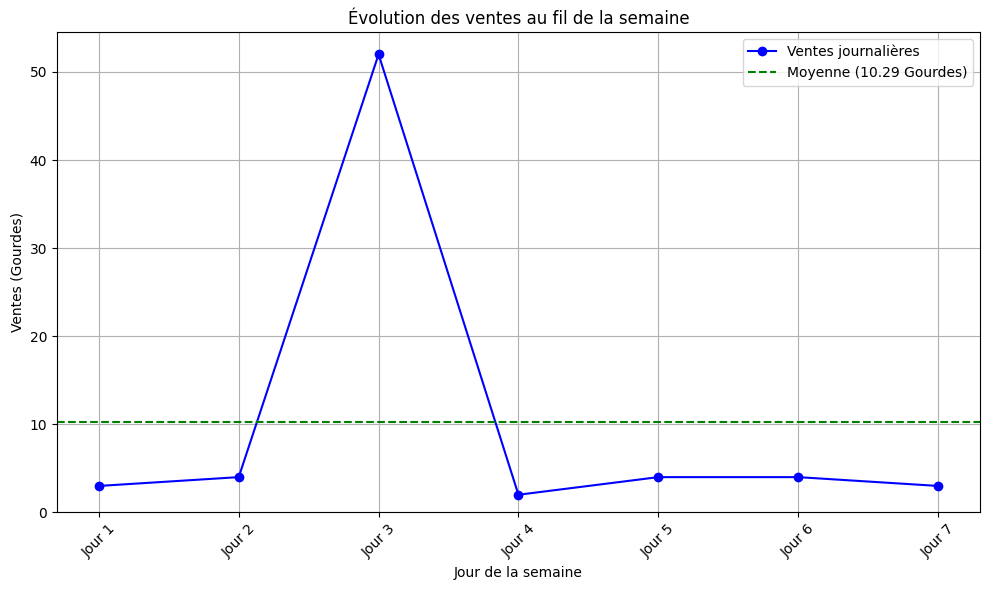
Veuillez entrer les ventes de votre magasin pour les 7 derniers jours.

Entrer la vente du jour 1 : 3  
Entrer la vente du jour 2 : 4  
Entrer la vente du jour 3 : 52  
Entrer la vente du jour 4 : 2  
Entrer la vente du jour 5 : 4  
Entrer la vente du jour 6 : 4  
Entrer la vente du jour 7 : q

Entrée invalide. Veuillez entrer un nombre pour les ventes.

Entrer la vente du jour 7 : 3

Le total des ventes est : 72.00 Gourdes  
La moyenne des ventes est : 10.29 Gourdes



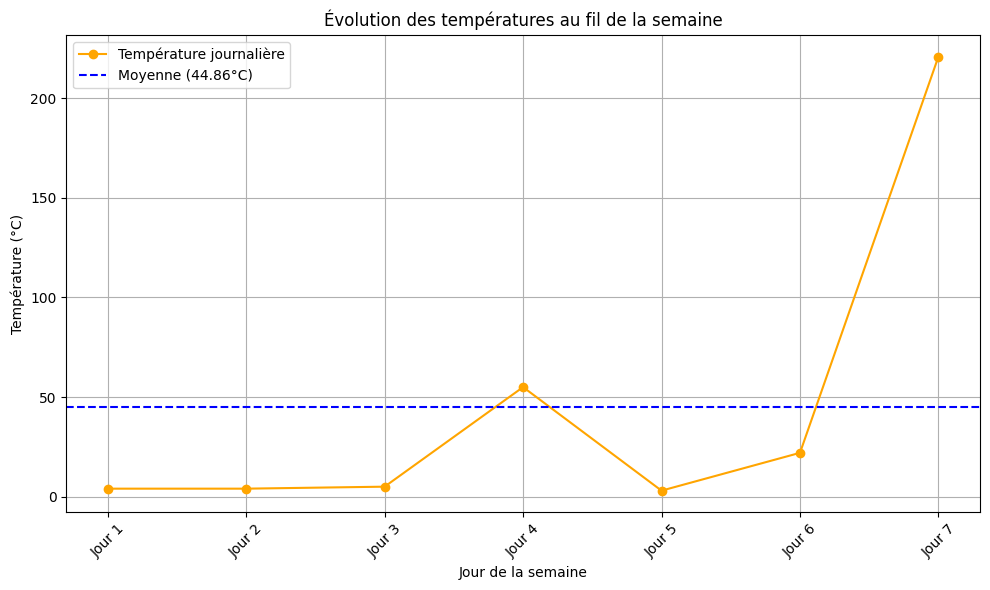
Demander à l’utilisateur d’entrer les températures journalières d’une semaine. Calculer la température moyenne. Afficher un graphe en ligne montrant l’évolution des températures.

import matplotlib.pyplot as plt  
  
# Liste pour stocker les températures journalières  
temperatures\_journalieres = []  
somme\_temperatures = 0  
  
print("Veuillez entrer les températures journalières pour la semaine.")  
  
for i in range(7):  
 while True:  
 try:  
 temp\_jour = float(input(f"Entrer la température du jour {i+1} (°C) : "))  
 temperatures\_journalieres.append(temp\_jour)  
 somme\_temperatures += temp\_jour  
 break  
 except ValueError:  
 print("Entrée invalide. Veuillez entrer un nombre pour la température.")  
  
# Calcul de la moyenne des températures  
if len(temperatures\_journalieres) > 0:  
 moyenne\_temperatures = somme\_temperatures / len(temperatures\_journalieres)  
else:  
 moyenne\_temperatures = 0 # Cas où aucune température n'est entrée  
  
print(f"\nLa température moyenne de la semaine est : {moyenne\_temperatures:.2f} °C")  
  
  
## Évolution des températures au fil de la semaine  
  
# Créer une liste de labels pour l'axe des x  
jours = [f"Jour {i+1}" for i in range(len(temperatures\_journalieres))]  
  
plt.figure(figsize=(10, 6))  
  
# Tracer l'évolution des températures  
plt.plot(jours, temperatures\_journalieres, marker='o', linestyle='-', color='orange', label='Température journalière')  
  
# Tracer la moyenne des températures  
plt.axhline(y=moyenne\_temperatures, color='blue', linestyle='--', label=f'Moyenne ({moyenne\_temperatures:.2f}°C)')  
  
plt.xlabel("Jour de la semaine")  
plt.ylabel("Température (°C)")  
plt.title("Évolution des températures au fil de la semaine")  
plt.legend()  
plt.grid(True)  
# plt.ylim n'est pas strictement nécessaire pour les températures si on veut voir la variation,  
# mais on peut l'ajouter si on veut fixer une échelle  
# plt.ylim(bottom=-10, top=40) # Exemple de limites si besoin  
plt.xticks(rotation=45)  
plt.tight\_layout()  
plt.show()

Veuillez entrer les températures journalières pour la semaine.

Entrer la température du jour 1 (°C) : 4  
Entrer la température du jour 2 (°C) : 4  
Entrer la température du jour 3 (°C) : 5  
Entrer la température du jour 4 (°C) : 55  
Entrer la température du jour 5 (°C) : 3  
Entrer la température du jour 6 (°C) : 22  
Entrer la température du jour 7 (°C) : 221

La température moyenne de la semaine est : 44.86 °C



exo 3 - Simuler un certain nombre de lancers de pièces (pile ou face). Enregistrer les résultats dans un fichier JSON. Afficher le nombre de fois où on obtient "Pile" et "Face".

import random  
import json  
  
num\_lancers = int(input("Entrez le nombre de lancers de pièces : "))  
resultats = []  
for \_ in range(num\_lancers):  
 resultat = random.choice(["Pile", "Face"])  
 resultats.append(resultat)  
  
with open("resultats\_lancers.json", "w") as f:  
 json.dump(resultats, f)  
  
num\_pile = resultats.count("Pile")  
num\_face = resultats.count("Face")  
  
print(f"Nombre de 'Pile' : {num\_pile}")  
print(f"Nombre de 'Face' : {num\_face}")

Entrez le nombre de lancers de pièces : 10

Nombre de 'Pile' : 7  
Nombre de 'Face' : 3

import matplotlib.pyplot as plt # Importe la bibliothèque pour les graphiques  
import math  
  
# --- Définition des fonctions de calcul de volume ---  
def volume\_sphere(radius):  
 """Calcule le volume d'une sphère."""  
 # Formule: V = 4/3 \* pi \* r^3  
 return (4/3) \* math.pi \* (radius \*\* 3)  
  
def volume\_prisme(length, width, height):  
 """Calcule le volume d'un prisme rectangulaire."""  
 # Formule: V = L \* l \* h  
 return length \* width \* height  
  
def volume\_cone(radius, height):  
 """Calcule le volume d'un cône."""  
 # Formule: V = 1/3 \* pi \* r^2 \* h  
 return (1/3) \* math.pi \* (radius \*\* 2) \* height  
# --- Fin de la définition des fonctions ---  
  
# Initialiser les volumes à 0.0 pour tous les solides.  
# Ces variables seront utilisées pour le graphique de comparaison.  
volume\_sphere\_final = 0.0  
volume\_prisme\_final = 0.0  
volume\_cone\_final = 0.0  
  
# --- Demander à l'utilisateur de choisir un solide ---  
print("Choisissez le solide dont vous souhaitez calculer le volume :")  
print("1. Sphère (V = 4/3 \* π \* r³)")  
print("2. Prisme Rectangulaire (V = L \* l \* h)")  
print("3. Cône (V = 1/3 \* π \* r² \* h)")  
  
choix\_solide = 0  
while True: # Boucle pour s'assurer d'une entrée valide pour le choix  
 try:  
 choix\_solide = int(input("Entrez votre choix (1, 2 ou 3) : "))  
 if 1 <= choix\_solide <= 3:  
 break # Sort de la boucle si le choix est valide  
 else:  
 print("Choix invalide. Veuillez entrer 1, 2 ou 3.")  
 except ValueError:  
 print("Entrée invalide. Veuillez entrer un nombre entier.")  
  
# --- Obtenir les dimensions nécessaires et calculer le volume du solide choisi ---  
volume\_solide\_choisi = 0.0 # Variable pour stocker le volume du solide qui a été choisi  
nom\_solide\_choisi = ""  
  
if choix\_solide == 1:  
 nom\_solide\_choisi = "Sphère"  
 print("\n--- Dimensions de la Sphère ---")  
 while True: # Boucle pour s'assurer d'une entrée valide pour le rayon  
 try:  
 rayon = float(input("Entrez le rayon de la sphère : "))  
 if rayon < 0:  
 print("Le rayon ne peut pas être négatif. Veuillez réessayer.")  
 else:  
 volume\_sphere\_final = volume\_sphere(rayon)  
 volume\_solide\_choisi = volume\_sphere\_final  
 break # Sort de la boucle si le rayon est valide  
 except ValueError:  
 print("Entrée invalide. Veuillez entrer un nombre pour le rayon.")  
elif choix\_solide == 2:  
 nom\_solide\_choisi = "Prisme Rectangulaire"  
 print("\n--- Dimensions du Prisme Rectangulaire ---")  
 while True: # Boucle pour s'assurer d'une entrée valide pour la longueur  
 try:  
 longueur = float(input("Entrez la longueur du prisme : "))  
 if longueur < 0:  
 print("La longueur ne peut pas être négative. Veuillez réessayer.")  
 else:  
 break  
 except ValueError:  
 print("Entrée invalide. Veuillez entrer un nombre pour la longueur.")  
 while True: # Boucle pour s'assurer d'une entrée valide pour la largeur  
 try:  
 largeur = float(input("Entrez la largeur du prisme : "))  
 if largeur < 0:  
 print("La largeur ne peut pas être négative. Veuillez réessayer.")  
 else:  
 break  
 except ValueError:  
 print("Entrée invalide. Veuillez entrer un nombre pour la largeur.")  
 while True: # Boucle pour s'assurer d'une entrée valide pour la hauteur  
 try:  
 hauteur = float(input("Entrez la hauteur du prisme : "))  
 if hauteur < 0:  
 print("La hauteur ne peut pas être négative. Veuillez réessayer.")  
 else:  
 break  
 except ValueError:  
 print("Entrée invalide. Veuillez entrer un nombre pour la hauteur.")  
 volume\_prisme\_final = volume\_prisme(longueur, largeur, hauteur)  
 volume\_solide\_choisi = volume\_prisme\_final  
elif choix\_solide == 3:  
 nom\_solide\_choisi = "Cône"  
 print("\n--- Dimensions du Cône ---")  
 while True: # Boucle pour s'assurer d'une entrée valide pour le rayon  
 try:  
 rayon = float(input("Entrez le rayon du cône : "))  
 if rayon < 0:  
 print("Le rayon ne peut pas être négatif. Veuillez réessayer.")  
 else:  
 break  
 except ValueError:  
 print("Entrée invalide. Veuillez entrer un nombre pour le rayon.")  
 while True: # Boucle pour s'assurer d'une entrée valide pour la hauteur  
 try:  
 hauteur = float(input("Entrez la hauteur du cône : "))  
 if hauteur < 0:  
 print("La hauteur ne peut pas être négative. Veuillez réessayer.")  
 else:  
 break  
 except ValueError:  
 print("Entrée invalide. Veuillez entrer un nombre pour la hauteur.")  
 volume\_cone\_final = volume\_cone(rayon, hauteur)  
 volume\_solide\_choisi = volume\_cone\_final  
else:  
 # Cette branche ne devrait normalement pas être atteinte grâce à la boucle de validation du choix  
 print("Une erreur inattendue est survenue. Le programme va se terminer.")  
 exit()  
  
# --- Afficher le volume du solide choisi ---  
print(f"\nLe volume du {nom\_solide\_choisi} est : {volume\_solide\_choisi:.2f}")  
  
# --- Préparer les données pour le graphique en barres ---  
# La liste des noms des solides pour l'axe X  
solides\_noms = ["Sphère", "Prisme Rectangulaire", "Cône"]  
# La liste des volumes pour l'axe Y. Contient le volume calculé et 0 pour les autres.  
volumes\_pour\_graph = [volume\_sphere\_final, volume\_prisme\_final, volume\_cone\_final]  
  
# --- Créer et afficher le graphique en barres comparant les volumes des solides ---  
plt.figure(figsize=(10, 7)) # Définit la taille de la fenêtre du graphique  
plt.bar(solides\_noms, volumes\_pour\_graph, color=['skyblue', 'lightcoral', 'lightgreen']) # Crée les barres  
plt.xlabel("Type de Solide") # Étiquette de l'axe X  
plt.ylabel("Volume") # Étiquette de l'axe Y  
plt.title("Comparaison des Volumes des Solides") # Titre du graphique  
plt.grid(axis='y', linestyle='--', alpha=0.7) # Ajoute une grille horizontale pour la lisibilité  
plt.show() # Affiche le graphique

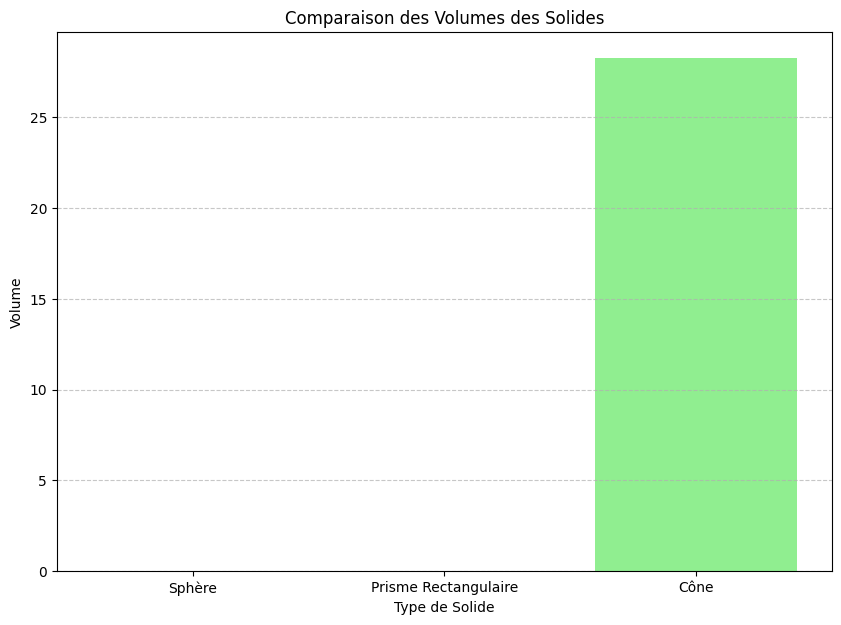
Choisissez le solide dont vous souhaitez calculer le volume :  
1. Sphère (V = 4/3 \* π \* r³)  
2. Prisme Rectangulaire (V = L \* l \* h)  
3. Cône (V = 1/3 \* π \* r² \* h)

Entrez votre choix (1, 2 ou 3) : 3

--- Dimensions du Cône ---

Entrez le rayon du cône : 3  
Entrez la hauteur du cône : 3

Le volume du Cône est : 28.27



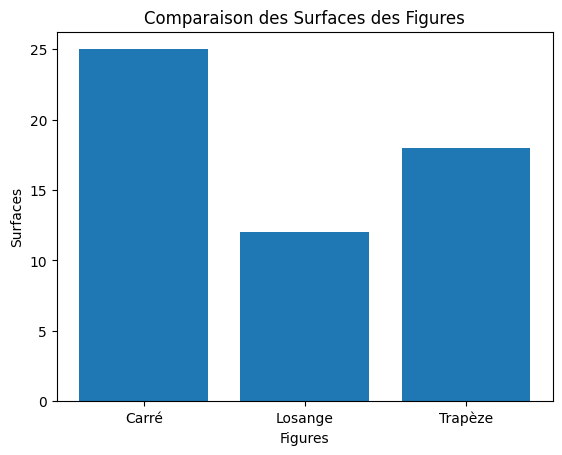
Exo 5 - Demander à l’utilisateur de choisir parmi trois figures : Carré → Périmètre : P=4c, Surface : S=c\*c Losange → Périmètre : P=4c, Surface : S=D×d/2 Trapèze → Périmètre : P=a+b+c+d, Surface : S=(B+b)×h/2 Calculer et afficher le périmètre et la surface de la figure choisie. Afficher un graphe comparant les surfaces des figures.

### import matplotlib.pyplot as plt # Présent dans plusieurs images  
  
# --- Définition des fonctions pour le périmètre et la surface des figures ---  
  
def carre\_perimetre(cote):  
 """Calcule le périmètre d'un carré."""  
 return 4 \* cote  
  
def carre\_surface(cote):  
 """Calcule la surface d'un carré."""  
 return cote \* cote  
  
def losange\_perimetre(cote):  
 """Calcule le périmètre d'un losange."""  
 return 4 \* cote  
  
def losange\_surface(grande\_diagonale, petite\_diagonale):  
 """Calcule la surface d'un losange."""  
 return (grande\_diagonale \* petite\_diagonale) / 2  
  
def trapeze\_perimetre(a, b, c, d):  
 """Calcule le périmètre d'un trapèze."""  
 return a + b + c + d  
  
def trapeze\_surface(grande\_base, petite\_base, hauteur):  
 """Calcule la surface d'un trapèze."""  
 return (grande\_base + petite\_base) \* hauteur / 2  
  
# --- Demander à l'utilisateur de choisir une figure ---  
print("Choisissez une figure :")  
print("1. Carré")  
print("2. Losange")  
print("3. Trapèze")  
  
choice = int(input("Entrez votre choix (1, 2 ou 3) : "))  
  
# --- Initialisation des variables pour éviter des erreurs si non définies ---  
perimetre = 0.0  
surface = 0.0  
  
# --- Obtenir les dimensions nécessaires en fonction du choix ---  
if choice == 1:  
 cote = float(input("Entrez la longueur du côté du carré : "))  
 perimetre = carre\_perimetre(cote)  
 surface = carre\_surface(cote)  
elif choice == 2:  
 cote = float(input("Entrez la longueur du côté du losange : ")) # Périmètre du losange  
 grande\_diagonale = float(input("Entrez la longueur de la grande diagonale du losange : "))  
 petite\_diagonale = float(input("Entrez la longueur de la petite diagonale du losange : "))  
 perimetre = losange\_perimetre(cote)  
 surface = losange\_surface(grande\_diagonale, petite\_diagonale)  
elif choice == 3:  
 a = float(input("Entrez la longueur du côté a du trapèze : "))  
 b = float(input("Entrez la longueur du côté b du trapèze : "))  
 c = float(input("Entrez la longueur du côté c du trapèze : "))  
 d = float(input("Entrez la longueur du côté d du trapèze : "))  
 grande\_base = float(input("Entrez la longueur de la grande base du trapèze : "))  
 petite\_base = float(input("Entrez la longueur de la petite base du trapèze : "))  
 hauteur = float(input("Entrez la hauteur du trapèze : "))  
 perimetre = trapeze\_perimetre(a, b, c, d)  
 surface = trapeze\_surface(grande\_base, petite\_base, hauteur)  
else:  
 print("Choix invalide.")  
 exit() # Quitte le programme si le choix est invalide  
  
# --- Afficher le périmètre et la surface de la figure choisie ---  
print(f"Le périmètre de la figure est : {perimetre}")  
print(f"La surface de la figure est : {surface}")  
  
# --- Données pour le graphique en barres ---  
figures = ["Carré", "Losange", "Trapèze"]  
# Les surfaces sont ici des exemples de valeurs comme dans l'image.  
# Si vous voulez que ces valeurs proviennent des calculs utilisateurs, il faudrait les stocker différemment.  
# Pour le moment, je garde les exemples de l'image.  
surfaces\_pour\_graphique = [carre\_surface(5), losange\_surface(6, 4), trapeze\_surface(7, 5, 3)] # Exemple de valeurs  
  
# --- Créer le graphique en barres ---  
plt.bar(figures, surfaces\_pour\_graphique)  
plt.xlabel("Figures")  
plt.ylabel("Surfaces")  
plt.title("Comparaison des Surfaces des Figures")  
plt.show()

Choisissez une figure :  
1. Carré  
2. Losange  
3. Trapèze

Entrez votre choix (1, 2 ou 3) : 3  
Entrez la longueur du côté a du trapèze : 3  
Entrez la longueur du côté b du trapèze : 1  
Entrez la longueur du côté c du trapèze : 6  
Entrez la longueur du côté d du trapèze : 2  
Entrez la longueur de la grande base du trapèze : 34  
Entrez la longueur de la petite base du trapèze : 12  
Entrez la hauteur du trapèze : 2

Le périmètre de la figure est : 12.0  
La surface de la figure est : 46.0



# Conclusion

En conclusion, ce TD me permet de mettre en œuvre des applications Python variées, me permet de développer des scripts pour calculer des totaux et des moyennes, de visualiser l'évolution de données (ventes, températures) avec des graphiques linéaires, et de créer des outils interactifs pour le calcul et la comparaison des volumes ou des surfaces de figures géométriques.