IUS

FST

TD2 - Maths-Info

Préparé par :

Pierre Yann Lelay

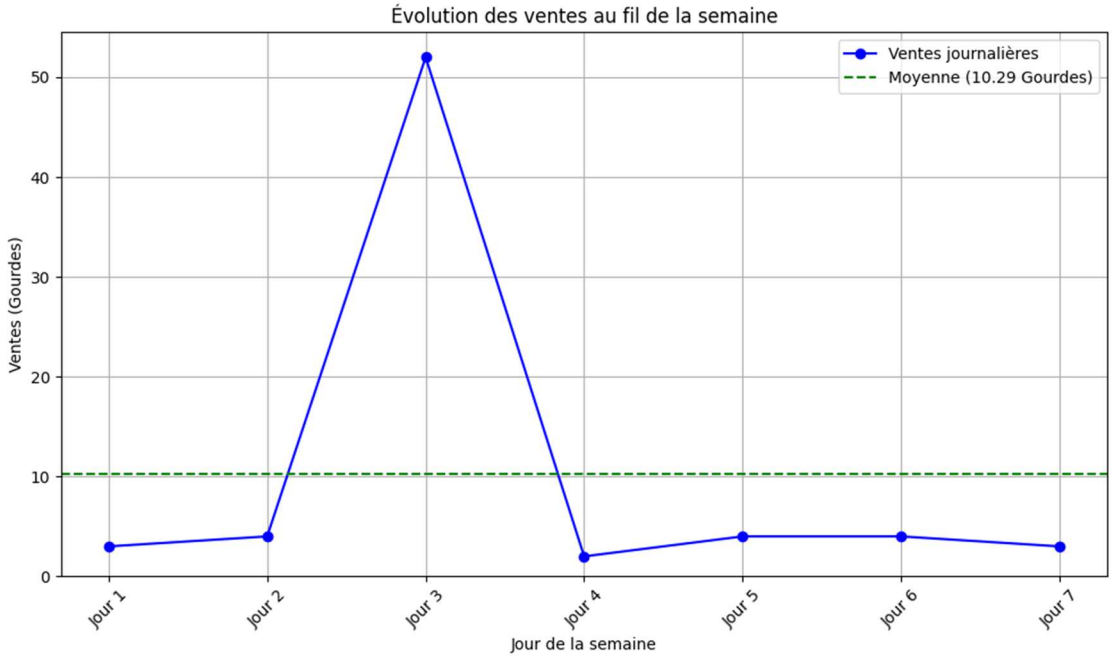
Etudiant : L3 -Sciences Informatiques

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
```



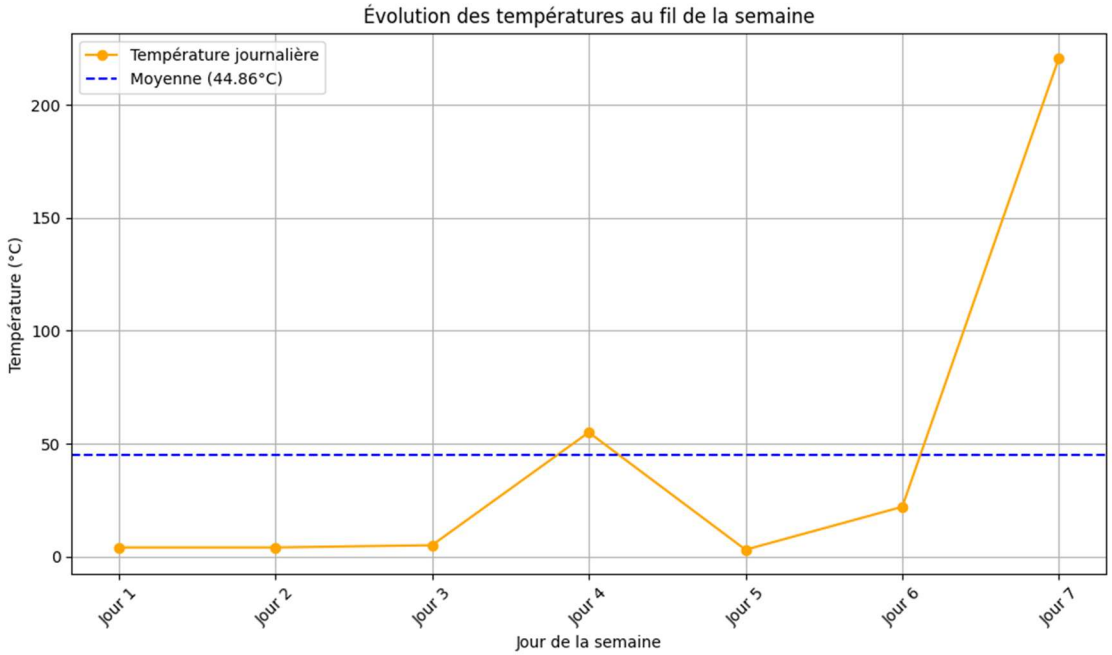
png

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
```



png

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
```

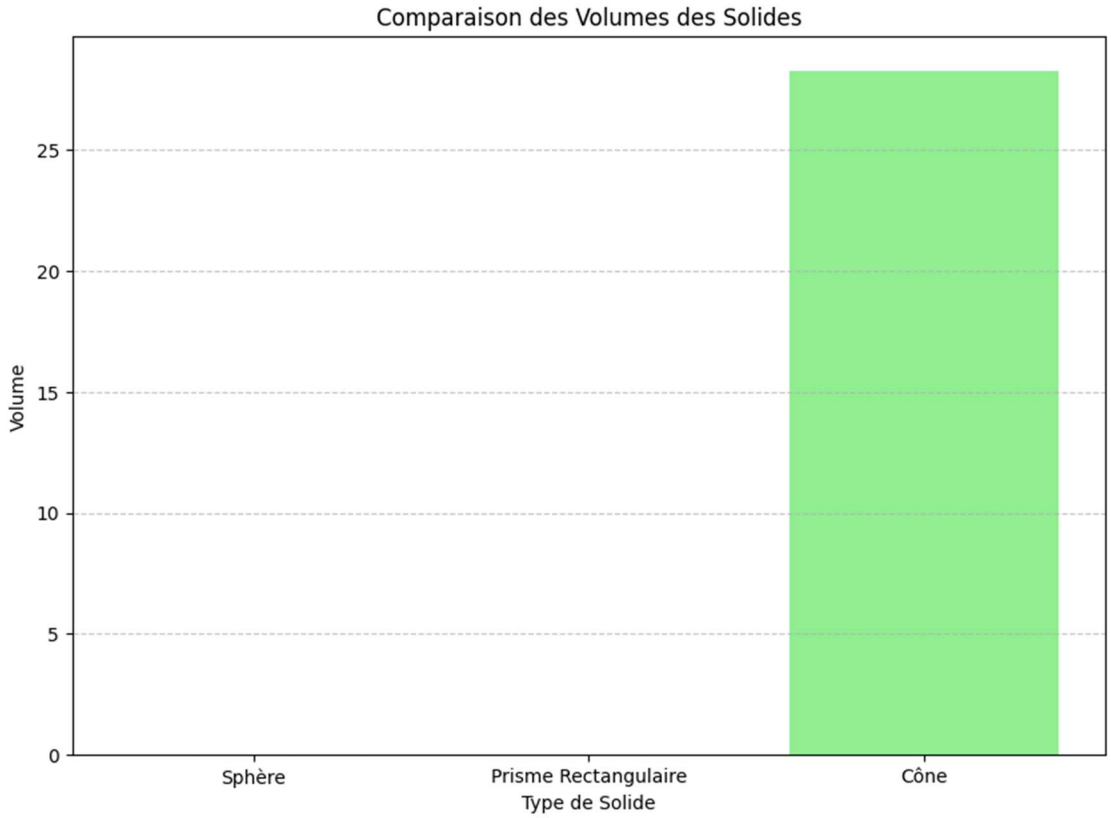
```python
import matplotlib.pyplot as plt # Importe la bibliothèque pour les graphiques
import math

# --- 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 = ""
```

```python
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
```



png

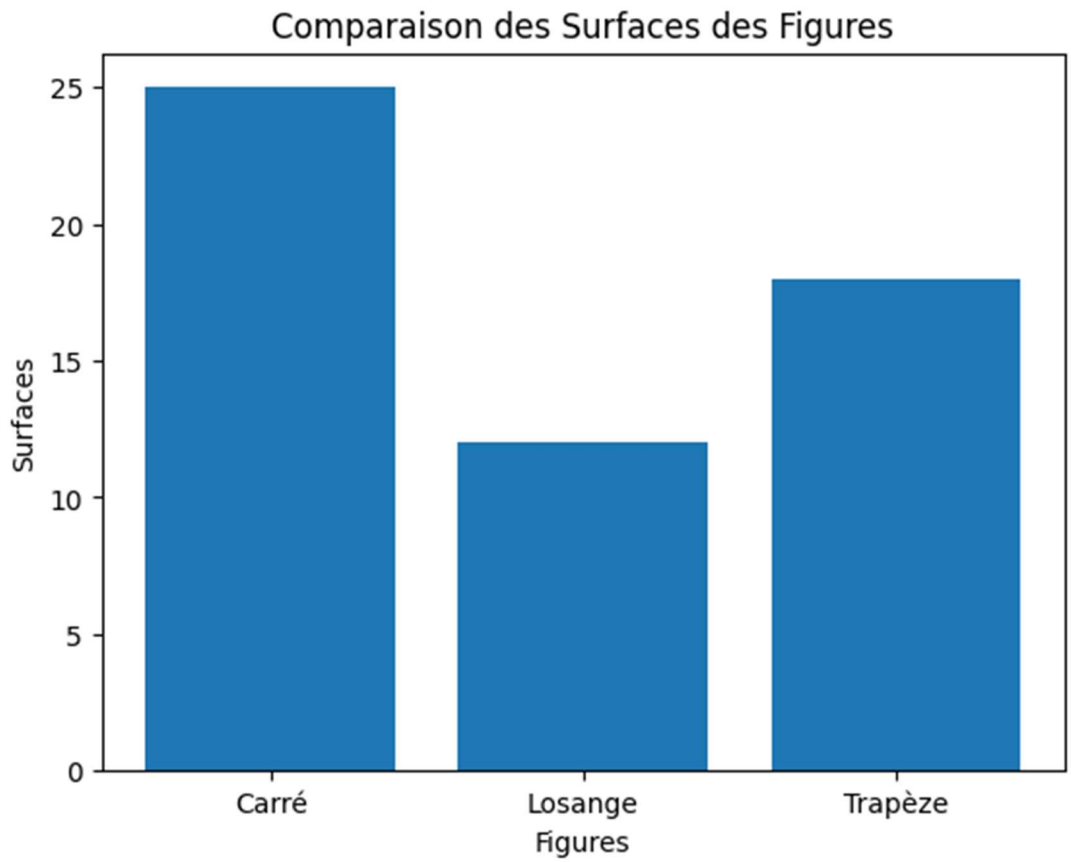
Exo 5 - Demander à l'utilisateur de choisir parmi trois figures : Carré → Périmètre : P=4c, Surface : S=c*c Losange → Périmètre : P=4c, Surface : S=D×d/2 Trapèze → Périmètre : P=a+b+c+d, Surface : S=(B+b)×h/2 Calculer et afficher le périmètre et la surface de la figure choisie. Afficher un graphe comparant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

```python
    """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
```



png

Conclusion

Tout au long de ces exercices, j'ai exploré et mis en œuvre des applications Python variées. J'ai développé des scripts pour calculer des totaux et des moyennes, visualiser l'évolution de données (ventes, températures) avec des graphiques linéaires, et même créer des outils interactifs pour le calcul et la comparaison des volumes ou des surfaces de figures géométriques. Chaque étape m'a permis de renforcer mes compétences en programmation Python, notamment en gestion des entrées utilisateur, en calculs mathématiques, en gestion des erreurs et en visualisation de données via la bibliothèque matplotlib