

```

# -*- coding: utf-8 -*-
"""
Created on Sun Jan  5 14:07:26 2025

@author: MohamedHamdani
"""

import pandas as pd
import numpy as np
import tkinter as tk
from tkinter import filedialog
import os

# Ouvrir une boîte de dialogue pour sélectionner le fichier Excel
root = tk.Tk()
root.withdraw()
file_path = filedialog.askopenfilename(title="Sélectionnez un fichier Excel", filetypes=[("Fichiers", "*.xlsx"), ("Fichiers", "*.xls")])

if not file_path:
    print("Aucun fichier sélectionné. Programme terminé.")
    exit()

try:
    # Charger le fichier Excel directement
    data = pd.ExcelFile(file_path)

    # Lire les feuilles et concaténer les données
    all_data = []
    for sheet in data.sheet_names:
        df = pd.read_excel(data, sheet_name=sheet)
        all_data.append(df)

    # Combiner toutes les feuilles
    combined_df = pd.concat(all_data)

    # Afficher les colonnes pour vérification
    print("Colonnes disponibles :", combined_df.columns.tolist())

    # Identifier les groupes de colonnes similaires (Date, H, Débit)
    column_sets = []
    for i in range(0, len(combined_df.columns), 3):
        cols = combined_df.columns[i:i+3]
        if len(cols) == 3 and 'Date' in cols[0] and 'H' in cols[1] and 'Débit' in cols[2]:
            column_sets.append(cols)

    # Vérifier si aucune colonne valide n'est détectée
    if not column_sets:
        raise KeyError("Colonnes requises non détectées dans le fichier.")

    # Créer une liste pour stocker les résultats
    results = []

    for cols in column_sets:
        temp_df = combined_df[cols].copy()
        temp_df.columns = ['Date', 'H', 'Débit'] # Renommer les colonnes

        # Nettoyer et convertir les valeurs

```

```

temp_df['Date'] = pd.to_datetime(temp_df['Date'], errors='coerce')
temp_df['Débit'] = pd.to_numeric(temp_df['Débit'], errors='coerce') # Convertir les valeurs

# Supprimer les lignes avec des valeurs manquantes
temp_df.dropna(subset=['Date', 'Débit'], inplace=True)

# Extraire l'année et le mois
temp_df['Année'] = temp_df['Date'].dt.year
temp_df['Mois'] = temp_df['Date'].dt.month

# Trouver les valeurs maximales de débit
max_debit = temp_df.loc[temp_df.groupby(['Année', 'Mois'])['Débit'].idxmax()]

# Ajouter les résultats
results.append(max_debit)

# Combiner les résultats
final_result = pd.concat(results).sort_values(['Année', 'Mois'])

# Enregistrer dans un fichier Excel
output_path = filedialog.asksaveasfilename(defaultextension=".xlsx", filetypes=[("Fichiers Excel", ".xlsx")])

if output_path:
    final_result.to_excel(output_path, index=False)
    print(f"Le fichier organisé a été enregistré sous : {output_path}")
else:
    print("Enregistrement annulé.")

except KeyError as e:
    print(f"Erreur : Colonne manquante ou invalide - {e}. Vérifiez le fichier source.")
except PermissionError:
    print("Erreur : Impossible d'accéder au fichier. Assurez-vous qu'il n'est pas ouvert dans une autre application.")
except Exception as e:
    print(f"Une erreur est survenue : {e}")

```