SAÉ Traiter les données

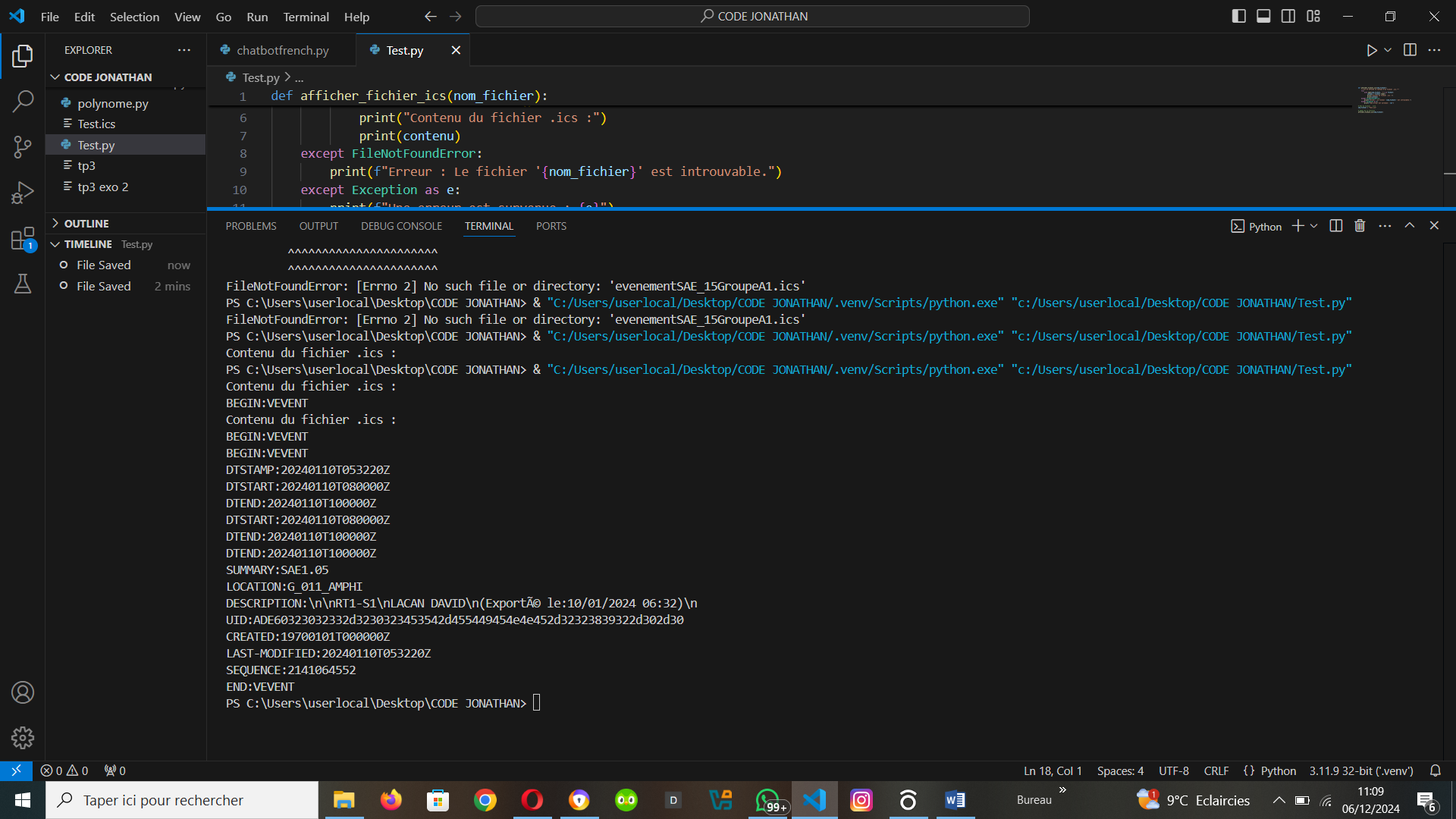
BEYEME EDOU Jenathan

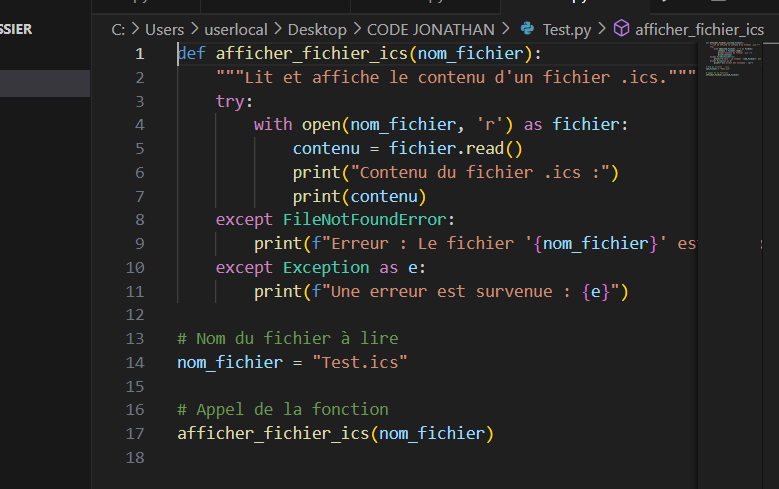
TP1 SAE105.

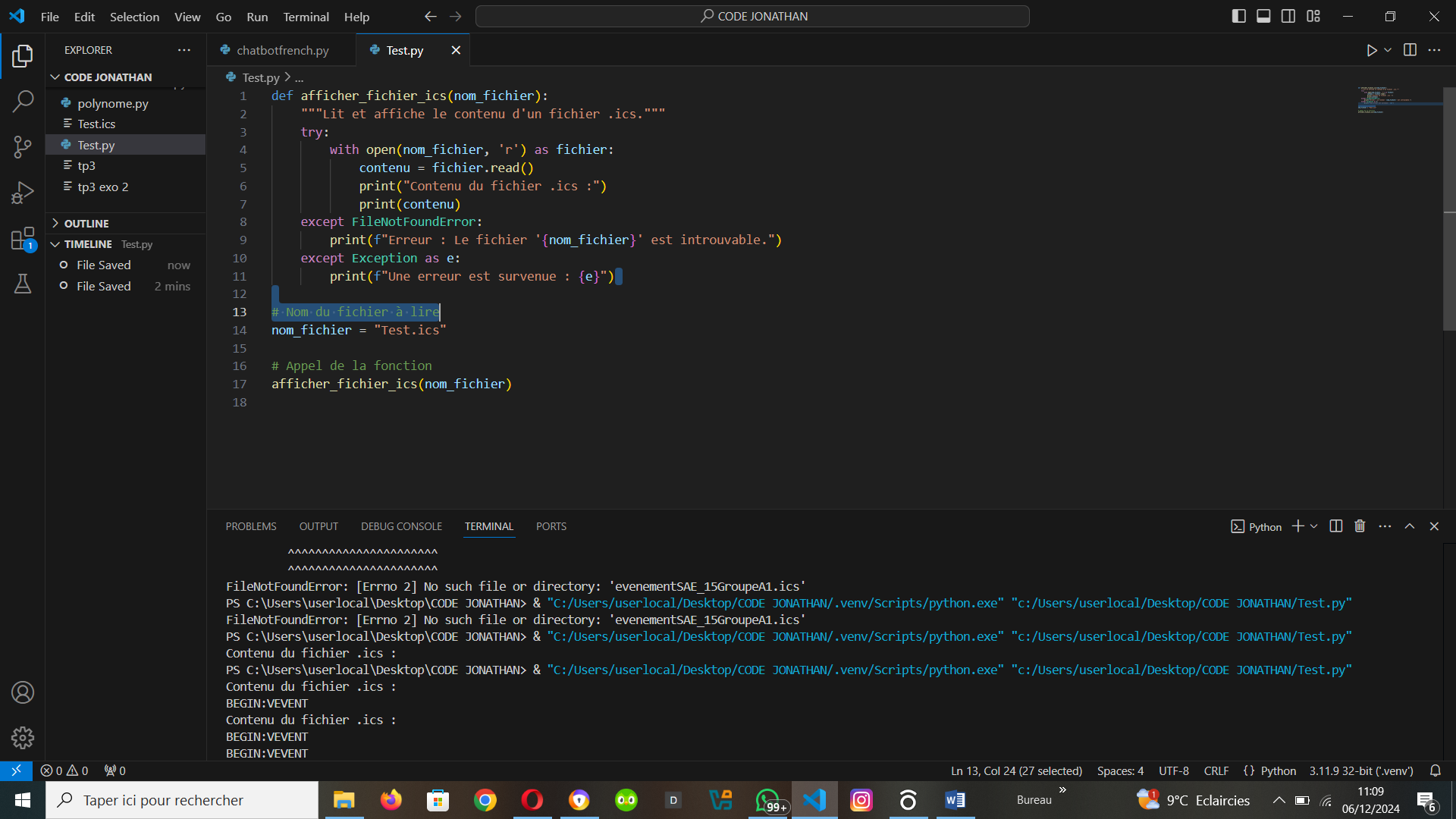
**Premier travaille :**

J’ai créé un programme appelé Test.py qui a réussi a afficher les données d’un fichier .ics

Où j’ai pu lire et afficher les données. J’ai enregistré ces fichiers dans le meme dossier « CODE JONATHAN »

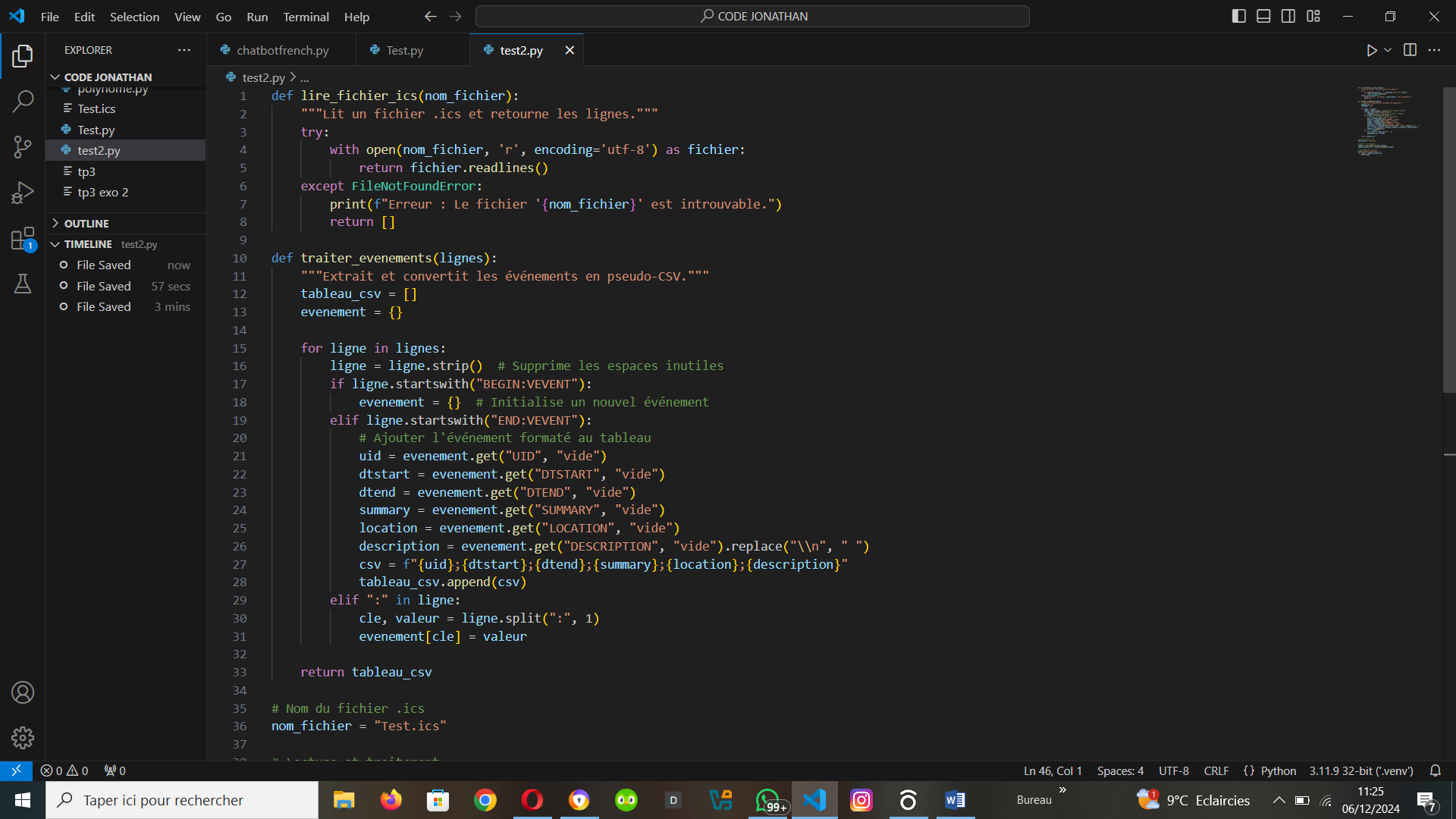


 Programme en python.

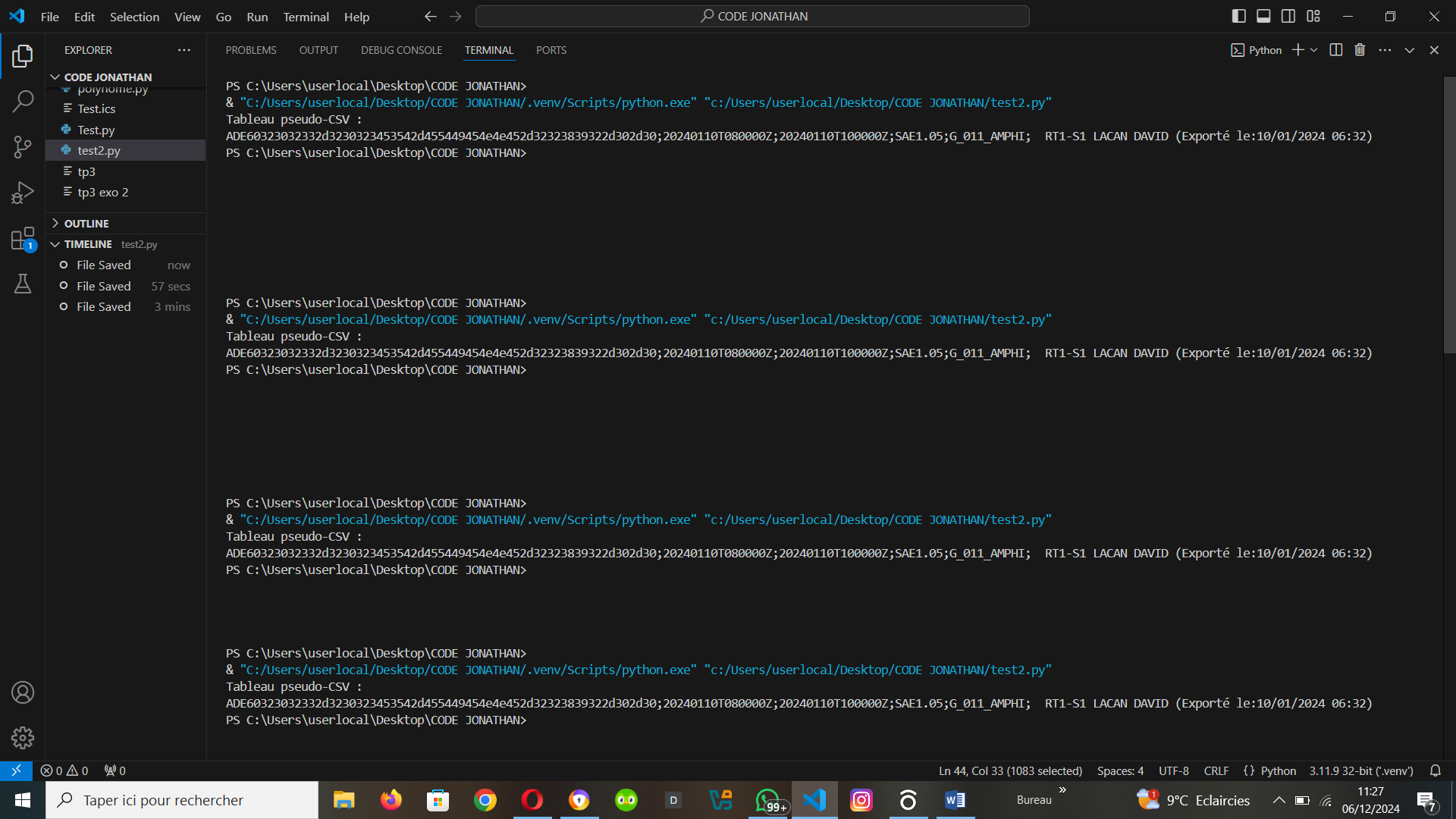


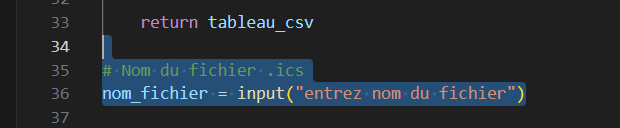
**Deuxième travail :**

Qui consiste à lister les données du fichier.ics sous fourme de tableau comportant les lignes du speudo code CV.



Le résultat.



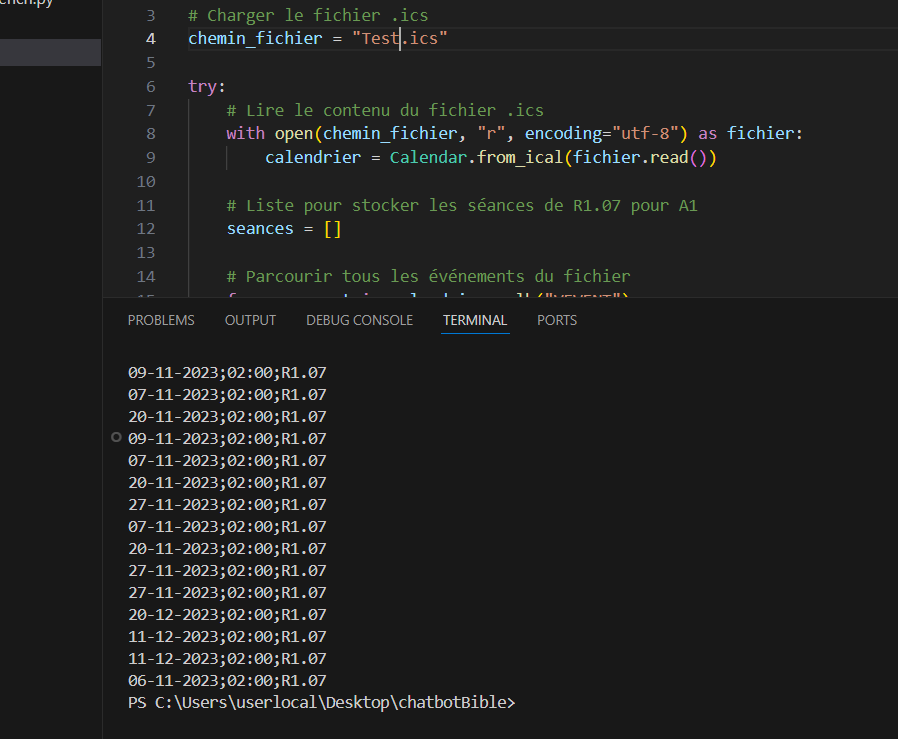
A l’intérieure de mon programme, comporte une fonction qui permet directement de prendre le fichier : 

**Troisième travail :**

Le **travail 3** vous demande d’extraire, à partir du tableau généré dans le **travail 2**, toutes les séances associées à la ressource **R1.07 (Informatique)** pour votre groupe de TP. Pour chaque séance, vous devez fournir un tableau contenant :

* **La date de la séance.**
* **La durée de la séance.**
* **Le type de séance**

Nous avons crée un script python qui nous a permis de lire tout en créant un tableau contenant les différentes valeur du fichier Test.ics, ce dernier contenait les jours, heures et séance de TP de R.107.

 voici le résultat du script.

Et voici le code qui nous a permis d’effectuer ce travaille.

from icalendar import Calendar

# Charger le fichier .ics

chemin\_fichier = "Test.ics"

try:

    # Lire le contenu du fichier .ics

    with open(chemin\_fichier, "r", encoding="utf-8") as fichier:

        calendrier = Calendar.from\_ical(fichier.read())

    # Liste pour stocker les séances de R1.07 pour A1

    seances = []

    # Parcourir tous les événements du fichier

    for evenement in calendrier.walk("VEVENT"):

        # Extraire les propriétés nécessaires

        description = evenement.get("DESCRIPTION", "")

        intitule = evenement.get("SUMMARY", "")

        date\_debut = evenement.decoded("DTSTART").strftime("%d-%m-%Y")

        duree = evenement.decoded("DTEND") - evenement.decoded("DTSTART")

        modalite = intitule.split(" ")[0]  # Type de séance (CM, TD, TP)

        # Vérifier si l'événement est pour R1.07 et le groupe A1

        if "R1.07" in intitule and "A1" in description:

            # Ajouter la séance dans le tableau

            heures, minutes = divmod(duree.total\_seconds() // 60, 60)

            duree\_formatee = f"{int(heures):02}:{int(minutes):02}"

            seances.append(f"{date\_debut};{duree\_formatee};{modalite}")

    # Afficher les séances filtrées

    if seances:

        print("Séances de R1.07 pour le groupe A1 :")

        for seance in seances:

            print(seance)

    else:

        print("Aucune séance trouvée pour R1.07 et le groupe A1.")

except FileNotFoundError:

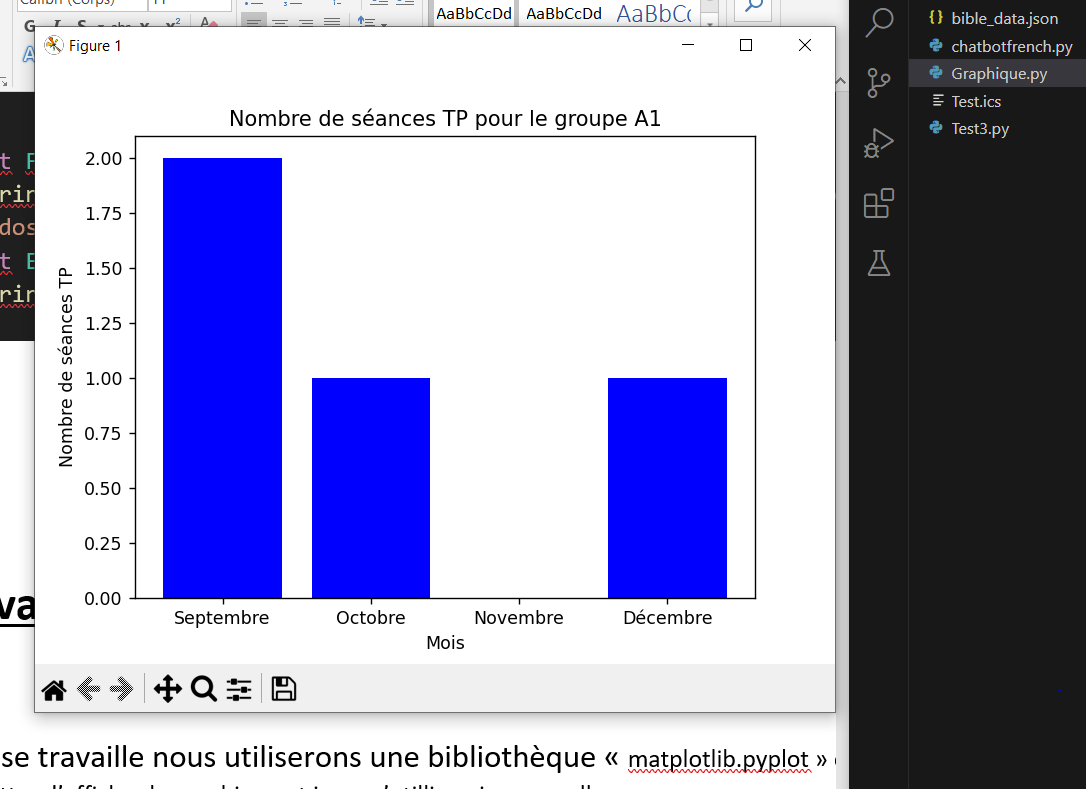
    print("Le fichier Test.ics est introuvable. Assurez-vous qu'il est dans le même dossier que ce script.")

except Exception as e:

    print(f"Une erreur est survenue : {e}")

**Travaille 4 :**

Pour se travaille nous utiliserons une bibliothèque « matplotlib.pyplot » qui nous permettra d’afficher le graphique et je ne n’utilliserai pas excell.

Voici le résultat de notre script qui avait pour but d’afficher un graphe qui présentait nos horaires de cours es mois antérieures.

import matplotlib.pyplot as plt

import csv

from datetime import datetime

# Supposons que vous ayez déjà un tableau 'evenements' avec vos données au format CSV

evenements = [

    # Exemple d'événements au format (date, type, groupe)

    ['10-09-2023', 'TP', 'A1'],

    ['12-09-2023', 'TP', 'A1'],

    ['05-10-2023', 'TP', 'A1'],

    ['19-11-2023', 'CM', 'A1'],

    ['25-12-2023', 'TP', 'A1']

]

# Initialiser un dictionnaire pour compter les TP par mois

tp\_par\_mois = {'09': 0, '10': 0, '11': 0, '12': 0}

# Filtrer et compter les événements TP pour le groupe A1

for evenement in evenements:

    date\_str = evenement[0]  # Exemple: '10-09-2023'

    evenement\_type = evenement[1]

    groupe = evenement[2]

    if evenement\_type == 'TP' and groupe == 'A1':

        date\_obj = datetime.strptime(date\_str, '%d-%m-%Y')

        mois = date\_obj.strftime('%m')

        if mois in tp\_par\_mois:

            tp\_par\_mois[mois] += 1

# Préparer les données pour le graphique

mois = ['Septembre', 'Octobre', 'Novembre', 'Décembre']

nombre\_tp = [tp\_par\_mois['09'], tp\_par\_mois['10'], tp\_par\_mois['11'], tp\_par\_mois['12']]

# Créer le graphique en bâtons

plt.bar(mois, nombre\_tp, color='blue')

# Ajouter des titres et des étiquettes

plt.title('Nombre de séances TP pour le groupe A1')

plt.xlabel('Mois')

plt.ylabel('Nombre de séances TP')

# Afficher le graphique

plt.show()

# Sauvegarder le graphique en PNG

plt.savefig('nombre\_seances\_tp.png')

**voici le programme qui nous a permis d’effectuer ce travaille.**

**Nb : travailler sur les programme qui appel les fichiers, connaitre comment utiliser excell**