Sae 1 .05

1.5 Premier travail à faire

commentaire de mon code :

-import numpy as np et import datetime : Ces lignes importent la bibliothèque NumPy sous le nom np et le module datetime.

-fh = open('C:/Users/amand/Music/Mon projet SAE15/TP1 SAE 1.5/PREMIER TRAVAIL/evenementSAE\_15.ics', 'r') : Cette ligne ouvre le fichier iCalendar spécifié en mode lecture et assigne l'objet fichier à la variable fh.

res = fh.read() : Cette ligne lit l'intégralité du contenu du fichier et le stocke dans la variable res.

fh.close() : Cette ligne ferme le fichier pour libérer les ressources système.

ress = res.split('\n') : Cette ligne divise le contenu du fichier (stocké dans res) en une liste de lignes en utilisant le caractère de saut de ligne ('\n') comme délimiteur. La liste résultante est stockée dans la variable ress.

print(ress) : Cette ligne affiche la liste des lignes obtenues à partir du fichier iCalendar.

2-5 Deuxième travail à faire

Bien sûr, je vais vous expliquer chaque ligne de code en détail :

```python

import re

import csv

```

- `import re`: Importe le module `re` (expression régulière) pour effectuer des opérations de recherche de motifs dans des chaînes de caractères.

- `import csv`: Importe le module `csv` pour faciliter la manipulation de fichiers CSV.

```python

def extract\_info(event):

"""

Fonction pour extraire les informations d'un événement.

"""

summary\_match = re.search(r'SUMMARY:(.\*)', event)

location\_match = re.search(r'LOCATION:(.\*)', event)

description\_match = re.search(r'DESCRIPTION:(.\*)', event)

summary = summary\_match.group(1).strip() if summary\_match else "vide"

location = location\_match.group(1).strip() if location\_match else "vide"

description = description\_match.group(1).strip() if description\_match else "vide"

return [summary, location, description]

```

- `def extract\_info(event):`: Définit une fonction nommée `extract\_info` qui prend en paramètre un événement iCalendar et extrait les informations importantes (titre, salle, description).

- `summary\_match = re.search(r'SUMMARY:(.\*)', event)`: Utilise une expression régulière pour rechercher une correspondance avec le motif "SUMMARY:" suivi de n'importe quel caractère (`.\*`) dans l'événement.

- `summary = summary\_match.group(1).strip() if summary\_match else "vide"`: Si une correspondance est trouvée, extrait le groupe capturé (le contenu du titre) et le nettoie (supprime les espaces indésirables). Sinon, attribue la chaîne "vide".

- Les mêmes principes sont appliqués aux lignes suivantes pour la recherche et l'extraction de la salle (`LOCATION`) et de la description (`DESCRIPTION`).

```python

def ics\_to\_csv(ics\_file, csv\_file):

"""

Fonction pour convertir un fichier iCalendar en pseudo-code CSV.

"""

with open(ics\_file, 'r') as file:

ics\_content = file.read()

events = ics\_content.split('BEGIN:VEVENT')[1:]

data = [['Titre', 'Salle', 'Description']]

```

- `def ics\_to\_csv(ics\_file, csv\_file):`: Définit une fonction `ics\_to\_csv` qui prend en paramètre le chemin d'un fichier iCalendar (`ics\_file`) et le chemin du fichier CSV résultant (`csv\_file`).

- `with open(ics\_file, 'r') as file: ...`: Utilise le contexte de gestion de fichiers pour ouvrir le fichier iCalendar en mode lecture (`'r'`). Le bloc `with` garantit que le fichier sera correctement fermé après utilisation.

- `ics\_content = file.read()`: Lit le contenu du fichier iCalendar.

```python

events = ics\_content.split('BEGIN:VEVENT')[1:]

data = [['Titre', 'Salle', 'Description']]

```

- `events = ics\_content.split('BEGIN:VEVENT')[1:]`: Divise le contenu en une liste d'événements en utilisant la chaîne 'BEGIN:VEVENT' comme séparateur. Le `[1:]` exclut la première partie qui n'est pas un événement.

- `data = [['Titre', 'Salle', 'Description']]`: Initialise une liste `data` avec les titres de colonnes pour le fichier CSV.

```python

for event in events:

event\_data = extract\_info(event)

data.append(event\_data)

with open(csv\_file, 'w', newline='', encoding='utf-8') as csvfile:

```

- `for event in events: ...`: Parcours chaque événement extrait précédemment.

- `event\_data = extract\_info(event)`: Appelle la fonction `extract\_info` pour obtenir les informations de l'événement.

- `data.append(event\_data)`: Ajoute les informations de l'événement à la liste `data`.

```python

csv\_writer = csv.writer(csvfile)

csv\_writer.writerows(data)

```

- `csv\_writer = csv.writer(csvfile)`: Initialise un objet `csv.writer` pour écrire dans le fichier CSV.

- `csv\_writer.writerows(data)`: Écrit toutes les lignes de données dans le fichier CSV.

```python

# Appel de la fonction pour convertir le fichier iCalendar en pseudo-code CSV

ics\_to\_csv('ADE\_RT1\_Septembre2023\_Decembre2023.ics', 'resultat\_pseudo\_code.csv')

```

- Appelle la fonction `ics\_to\_csv` avec le chemin du fichier iCalendar en entrée et le chemin du fichier CSV en sortie.

3-1/Troisième travail à faire