

Il file `storico.csv` contiene i dati giornalieri di temperatura registrati nei capoluoghi di provincia del Friuli Venezia Giulia durante il 2023, in formato CSV:

1: File CSV: `storico.csv`

```
Trieste,Udine,Pordenone,Gorizia
3.4,-2.9,2.3,-2.1
8.6,-2.4,2.4,0.1
...
3.9,0.3,-4.2,2.0
```

Oltre alla riga di intestazione, contenente i nomi delle città, il file contiene 365 righe, una per ogni giorno dell'anno, dal 1 Gennaio al 31 Dicembre. Le temperature possono essere rappresentate in Python come una matrice (lista di liste di float) di dimensione 365×4 .

```
temperature_fvg = [[3.4, -2.9, 2.3, -2.1], [8.6, -2.4, 2.4, 0.1], ..., [3.9, 0.3, -4.2, 2.0]]
```

I nomi delle città possono essere rappresentati in Python come una lista di stringhe:

```
citta_fvg = ['Trieste', 'Udine', 'Pordenone', 'Gorizia']
```

Un fenomeno meteorologico rilevante è la cosiddetta ondata di calore: essa è definita come un periodo di tempo durante il quale si misura, per almeno 3 giorni consecutivi, una temperatura insolitamente elevata, uguale o superiore ad un determinato valore di soglia.

Si consideri, a titolo di esempio la seguente lista `t_esempio` con 13 valori giornalieri di temperatura.

```
t_esempio = [30.1, 32.1, 33.6, 34.2, 30.1, 30.4, 29.8, 31.4, 31.2, 29.2, 31.4, 31.7, 32.0]
```

Considerato un valore di soglia pari a 30, si possono identificare due ondate di calore, in corrispondenza dei valori evidenziati in grassetto.

Infine, si consideri che per lo svolgimento dell'esercizio verrà utilizzato il seguente dizionario che ha come chiavi i nomi dei mesi e come valori il numero di giorni ad essi associati.

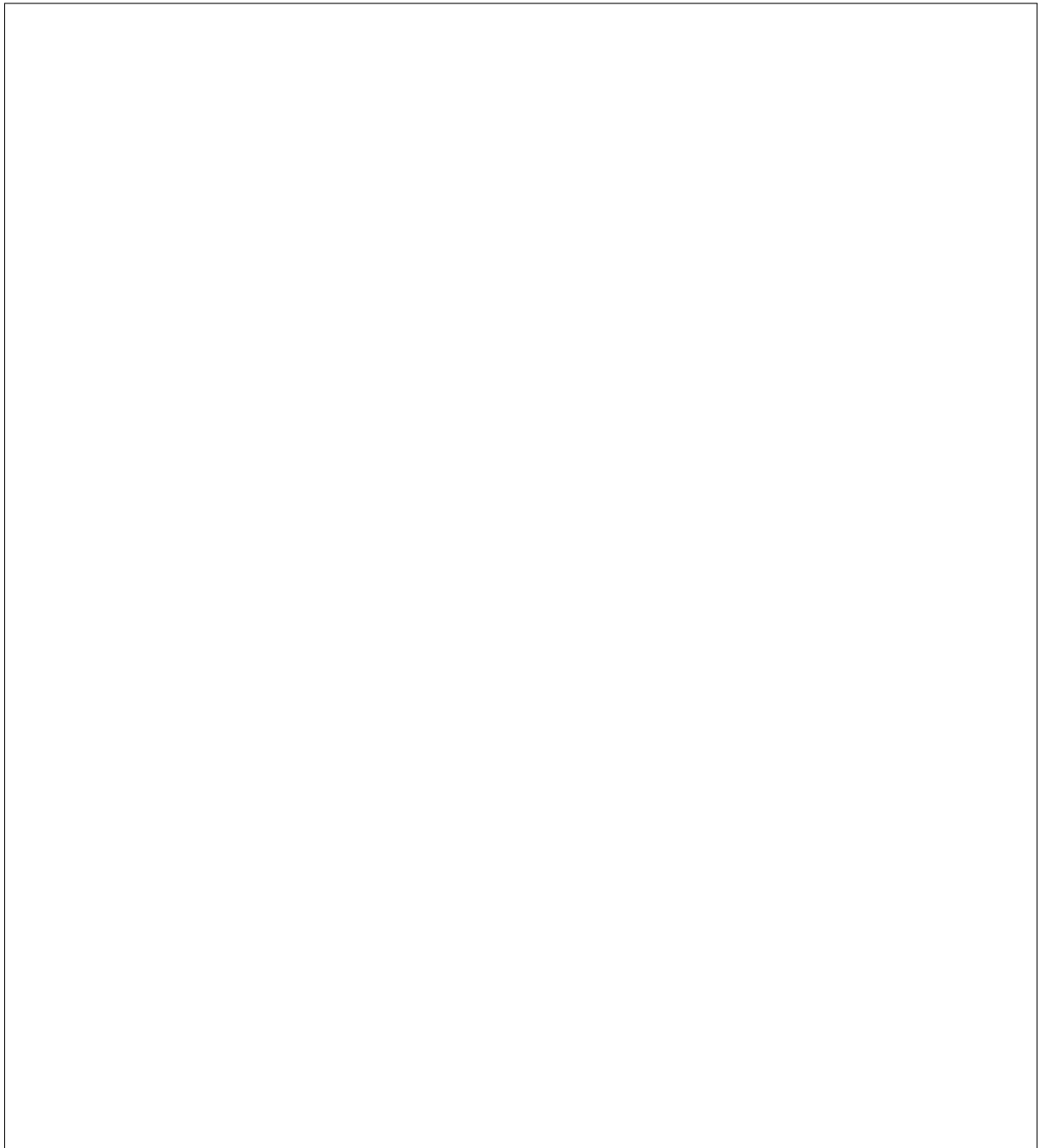
```
giorni_mese = { 'Gennaio': 31, 'Febbraio': 28, 'Marzo': 31, 'Aprile': 30,
                'Maggio': 31, 'Giugno': 30, 'Luglio': 31, 'Agosto': 31, 'Settembre': 30,
                'Ottobre': 31, 'Novembre': 30, 'Dicembre': 31 }
```

Nell'ultima pagina è fornita l'implementazione (parziale) della funzione `main`: sono già presenti gli statement Python per la definizione di tale dizionario e per la lettura del contenuto del file `storico.csv`.

EDIT: Il file `storico.csv` è disponibile a questo link
https://drive.google.com/file/d/1aJVgIgCLeNLavbVXz0h0_rTIZleAsA-M/view?usp=sharing

1. Definire la funzione `trova_picco_calore`

- **Parametri di ingresso:**
 - `temperature`: lista di liste di float.
- **Restituisce:** una tupla costituita da due interi.
- **Descrizione:** la funzione individua il valore massimo all'interno della matrice `temperature` fornita in ingresso. Quindi, restituisce una tupla contenente gli indici di riga e di colonna relativi a una delle occorrenze del valore massimo (nel caso in cui ce ne siano più di una, è sufficiente restituirne una sola).
- **Output atteso:** A titolo di esempio, si consideri la variabile `temperature` come riportata nella pagina iniziale e si considerino solo i 12 valori rappresentati. La funzione restituisce (1, 0), tupla degli indici di riga e colonna relativi al valore massimo pari a 8.6.



2. Definire la funzione `ottieni_data`

- **Parametri di ingresso:**

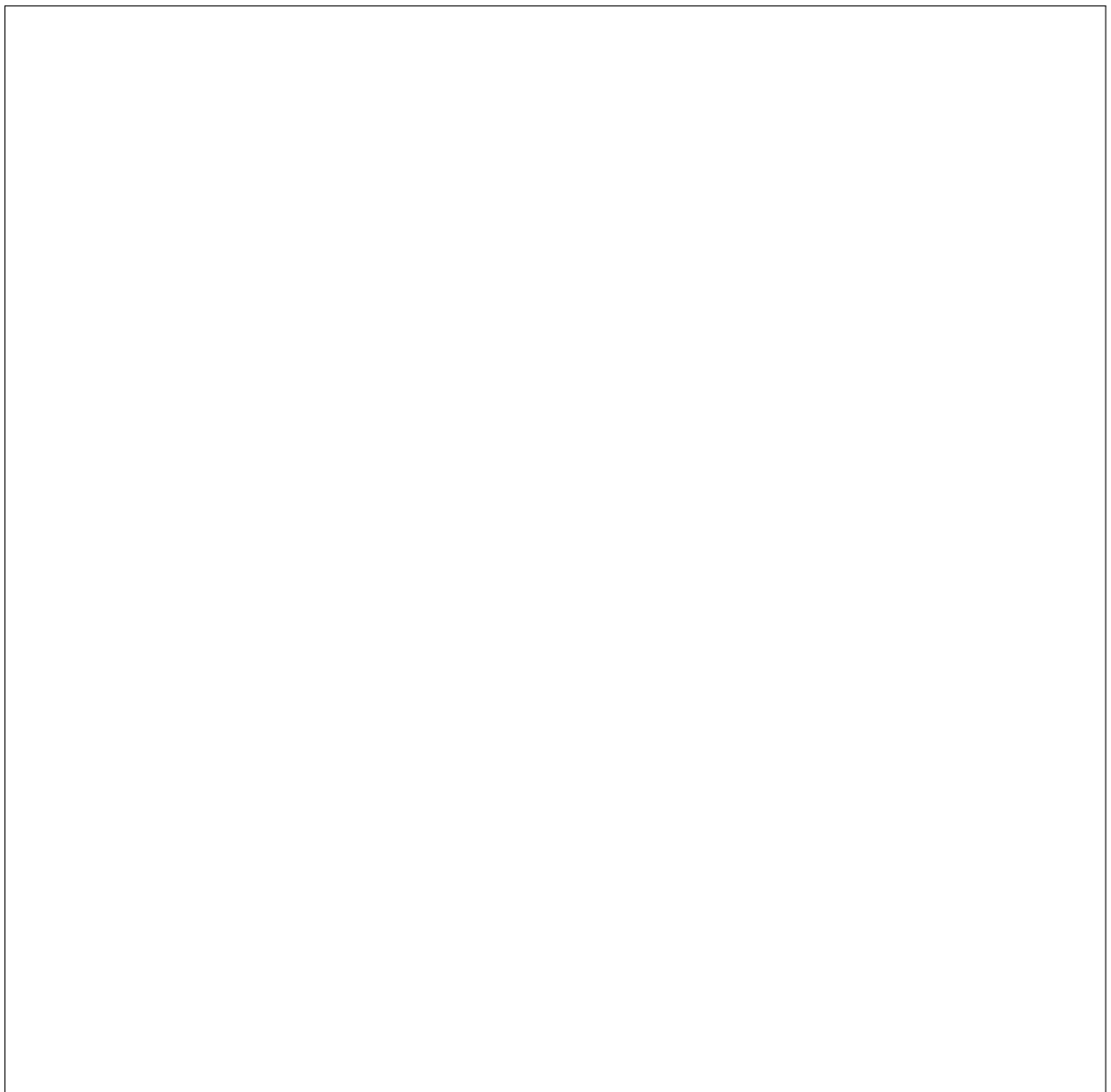
- `indice_giorno`: un intero compreso tra 0 e 364.
- `giorni_mese`: il dizionario che associa a ciascun nome del mese il relativo numero di giorni.

- **Restituisce:** una stringa.

- **Descrizione:** la funzione restituisce una stringa che rappresenta la data (giorno e mese) corrispondente al valore di `indice_giorno`. La valutazione sfrutta il dizionario `giorni_mese` come definito nella descrizione iniziale.

- **Output atteso:**

- `indice_giorno = 0` → la funzione restituisce `'1 Gennaio'`
- `indice_giorno = 31` → la funzione restituisce `'1 Febbraio'`
- `indice_giorno = 364` → la funzione restituisce `'31 Dicembre'`



3. Definire la funzione `conta_ondate_calore`

- **Parametri di ingresso:**
 - `temperature`: una lista di liste di float.
 - `lista_citta`: una lista di stringhe.
 - `soglia`: un intero, con valore di default pari a 30.
- **Restituisce:** un dizionario che associa una stringa (chiave) ad un intero (valore).
- **Descrizione:** la funzione restituisce un dizionario che associa a ciascuna stringa della lista `lista_citta` un intero relativo alla corrispondente colonna della matrice `temperature`. Data una singola colonna, tale intero è calcolato come il numero di sequenze di almeno 3 valori consecutivi uguali o superiori alla soglia.
- **Output atteso:** Con riferimento ai capoluoghi di provincia del FVG, la funzione restituisce un dizionario che associa a ciascuna città il numero di ondate di calore registrate (si noti che i valori sono solo esemplificativi):

```
{ 'Trieste': 2, 'Udine': 3, 'Pordenone': 1, 'Gorizia': 2 }
```

4. Infine, completare la funzione `main` in modo che svolga i seguenti compiti:

- ottenga gli indici di riga e colonna relativi al picco di calore, ovvero il valore massimo all'interno della matrice `temperature_fvg`, utilizzando la funzione `trova_picco_calore`.
- stampi la città, il giorno e il mese in cui si è verificato il picco di calore. Utilizzare la funzione `ottieni_data` per la conversione dell'indice nel formato richiesto. Esempio:
`'Picco di calore rilevato a Trieste il 22 Luglio'`
- stampi riga per riga il numero di ondate di calore rilevate in ciascuna città, calcolati utilizzando la funzione `conta_ondate_calore` ed opportunamente formattati. Esempio di output atteso:

Trieste	2
Udine	3
Pordenone	1
Gorizia	2

```
import csv

def main():
    giorni_mese = {
        'Gennaio': 31, 'Febbraio': 28, 'Marzo': 31, 'Aprile': 30, 'Maggio': 31,
        'Giugno': 30, 'Luglio': 31, 'Agosto': 31, 'Settembre': 30, 'Ottobre': 31,
        'Novembre': 30, 'Dicembre': 31
    }

    with open('storico.csv', 'r') as mf:
        csvReader = csv.reader(mf)
        citta_fvg = next(csvReader)
        temperature_fvg = []
        for linea in csvReader:
            temperature_fvg.append([float(x) for x in linea])

    # citta_fvg e' una lista di stringhe contenente i nomi delle quattro citta'
    # temperature_fvg e' una matrice 365 x 4 (lista di liste di float)
    # contenente lo storico delle temperature per le quattro citta'
```