```python
import requests #Library required for using APIs

def getWeather(latitude, longitude):
    """
    Fetches the current weather for a given latitude and longitude.

    Parameters:
    latitude (float): The latitude of the location.
    longitude (float): The longitude of the location.

    Returns:
    tuple: A tuple containing the temperature (float) and weather code (int) if the request is successful, otherwise None.
    """
    #Set up the API request URL
    url = f"https://api.open-meteo.com/v1/forecast?latitude={latitude}&longitude={longitude}&current_weather=true"
    response = requests.get(url)
    if response.status_code == 200: #Check if the request was successful
        #Extract the temperature and weather code from the API response
        data = response.json()
        temperature = data['current_weather']['temperature']
        weatherCode = data['current_weather']['weathercode']
        return temperature, weatherCode
    else:
        return None

```

```python
def displayWeather(latitude, longitude):
    """
    Fetches and displays the weather information for a given latitude and longitude.

    Args:
        latitude (float): The latitude of the location.
        longitude (float): The longitude of the location.

    Returns:
        None

    Prints:
        The temperature in Celsius and a description of the weather condition.
        If the API request fails, prints "API request failed".
    """
    temperature, weatherCode = getWeather(latitude, longitude) #Get the temperature and weather code at a certain location
    weatherDict = {
        0: "Clear skies",
        1: "Mostly clear",
        2: "Partly cloudy",
        3: "Overcast",
        45: "Foggy",
        48: "Depositing rime fog",
        51: "Light drizzle",
        53: "Moderate drizzle",
        55: "Heavy drizzle",
        56: "Light freezing drizzle",
        57: "Heavy freezing drizzle",
        61: "Light rain",
        63: "Moderate rain",
        65: "Heavy rain",
        66: "Light freezing rain",
        67: "Heavy freezing rain",
        71: "Light snow fall",
        73: "Moderate snow",
        75: "Heavy snow",
        77: "Hail",
        80: "Light showers",
        81: "Moderate showers",
        82: "Violent rain",
        85: "Light snow",
        86: "Heavy snow",
        95: "Light thunderstorm",
        96: "Moderate thunderstorm",
        99: "Heavy thunderstorm"
    }
    if temperature is not None:
        print(f"{temperature}°C, {weatherDict[weatherCode]}")
    else:
        print("API request failed")

```

```python
def getCoords(city):
    """
    Retrieves the latitude and longitude coordinates for a given city using the Open-Meteo Geocoding API.

    Args:
        city (str): The name of the city for which to retrieve coordinates.

    Returns:
        tuple: A tuple containing the latitude and longitude of the city if found, otherwise (None, None).
    """
    #Set up the API request URL
    url = f"https://geocoding-api.open-meteo.com/v1/search?name={city}"
    response = requests.get(url)
    if response.status_code == 200: #Check if API request was successful
        #Extract latitude and longitude from the API response
        data = response.json()
        if data['results']:
            latitude = data['results'][0]['latitude']
            longitude = data['results'][0]['longitude']
            return latitude, longitude
        else:
            return None, None
    else:
        return None, None

def main():
    print("Welcome to the Weather App!")
    print("Enter the name of a city to get the current weather.")
    cont = True
    while cont:
        cityName = input("Enter city name: ")
        latitude, longitude = getCoords(cityName)
        if latitude is not None:
            displayWeather(latitude, longitude)
        else:
            print("City not found")
        cont = input("Do you want to check the weather for another city? (y/n) ") == "y"

if __name__ == "__main__":
    main()
```