```
#!/usr/bin/env python

import os
import sys
import requests
from datetime import datetime

# Global Parameters
LANG = 'en'

# DarkSky.net API Parameters
DS_API_HOST = 'https://api.darksky.net/forecast'
DS_API_KEY = os.environ.get('DS_API_KEY')
DS_UNITS = 'si'

# Google Maps Geocoding API Parameters
GM_ENDPOINT = 'http://maps.google.com/maps/api/geocode/json'
GM_API_KEY = os.environ.get('GM_API_KEY')

def make_get_request(uri: str, payload):
    """
    Function to make a GET request to API Endpoint
    :param uri:
    :param payload:
    :return:
    """
    response = requests.get(uri, payload)
    if response.status_code != 200:
        return None
    else:
        return response.json()

def get_geo_data(address: str):
    """ Function to get coordinates from Google Maps Geocoding API
```

```
    :param address:
    :return:
    """
    payload = {'address': address, 'language': LANG, 'key': GM_API_KEY}
    response = make_get_request(GM_ENDPOINT, payload)

    if not response:
        return None

    data = response['results'][0]
    formatted_address = data['formatted_address']
    lat = data['geometry']['location']['lat']
    lng = data['geometry']['location']['lng']

    return {'lat': lat, 'lng': lng, 'formatted_address': formatted_address}

def get_forecast_data(lat: str, lng: str):
    """ Function to get Forecast data from DarkSky.net API
    :param lat:
    :param lng:
    :return:
    """
    uri = DS_API_HOST + '/' + DS_API_KEY + '/' + str(lat) + ',' + str(lng)
    payload = {'lang': LANG, 'units': DS_UNITS}
    response = make_get_request(uri, payload)

    if not response:
        return None

    return response['daily']

def print_daily_forecast(geo, forecast):
    """
    Function to print daily weather forecast information
```

```
    :param geo:
    :param forecast:
    """
    print('Getting Forecast for: ' + geo['formatted_address'])
    print('Weekly Summary: ' + forecast['summary'])
    print()

    for day in forecast['data']:
        date = datetime.fromtimestamp(day['time'])
        if date.date() == datetime.now().date():
            day_name = 'Today'
        else:
            day_name = date.strftime("%A")
        summary = day['summary']
        temperature_min = str(round(day['temperatureMin'])) + 'ºC'
        temperature_max = str(round(day['temperatureMax'])) + 'ºC'
        print(
            date.strftime('%d/%m/%Y') + ' (' + day_name + '): ' +
            summary + ' ' + temperature_min + ' - ' + temperature_max
        )
        print()

def print_header():
    print('---------------------------------')
    print('     WEATHER FORECAST 1.O       ')
    print('---------------------------------')
    print()

def main():
    """
    Main Function
    """
    if len(sys.argv) < 2 or DS_API_KEY is None:
        exit('Error: no location or env vars found')
```

```
    geo_data = get_geo_data(sys.argv[1])

    if not geo_data:
        exit('Error: Address not found or invalid response')

    forecast_data = get_forecast_data(geo_data['lat'], geo_data['lng'])

    if not forecast_data:
        exit('Error: Forecast not found or invalid response')

    # Print Output Forecast information
    print_header()
    print_daily_forecast(geo_data, forecast_data)

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