Asynchronous Programming and Multi-threading - Task 7

Program that reads city names from cities.csv, makes asynchronous weather API calls and outputs results in JSON format to results.json :

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Net.Http;
using System.Text.Json;
using System.Threading;
using System.Threading.Tasks;

public class WeatherData
{
   public string City { get; set; }
   public string Weather { get; set; }
}

class Program
{
   private static readonly SemaphoreSlim _semaphore = new SemaphoreSlim(5); // Max 5
concurrent requests
   private static readonly HttpClient _httpClient = new HttpClient();
   private static readonly string _apiKey = "22a132e549e553bc2c346a181276e463"; //
Replace with your OpenWeather API key

   static async Task Main()
   {
      try
      {
         // Add User-Agent header to prevent 401 Unauthorized
         _httpClient.DefaultRequestHeaders.UserAgent.ParseAdd("Mozilla/5.0");

         // Check if cities.csv exists
         if (!File.Exists("cities.csv"))
         {
            Console.WriteLine("Error: cities.csv file not found");
            return;
         }

         // Read city list
         var cities = await File.ReadAllLinesAsync("cities.csv");

         // Check if cities.csv is empty
```

```
        if (cities.Length == 0)
        {
            Console.WriteLine("Error: cities.csv file is empty");
            return;
        }

        var tasks = new List<Task<WeatherData>>();

        // Schedule async fetches for each city
        foreach (var city in cities)
        {
            tasks.Add(FetchWeatherAsync(city.Trim())); // Trim to avoid trailing spaces
        }

        // Wait for all tasks to complete
        var results = await Task.WhenAll(tasks);

        Console.WriteLine($"Retrieved weather data for {results.Length} cities");

        // Serialize results to JSON
        var jsonOutput = JsonSerializer.Serialize(results, new JsonSerializerOptions {
WriteIndented = true });
        await File.WriteAllTextAsync("results.json", jsonOutput);

        Console.WriteLine("Weather data saved to results.json");
    }
    catch (Exception ex)
    {
        Console.WriteLine($"Error in Main: {ex.Message}");
    }
}

static async Task<WeatherData> FetchWeatherAsync(string city)
{
    await _semaphore.WaitAsync();
    try
    {
        int retries = 3;
        int delay = 1000;

        for (int attempt = 1; attempt <= retries; attempt++)
        {
            try
            {
                var url =
$"https://api.openweathermap.org/data/2.5/weather?q={Uri.EscapeDataString(city)}&appid
={_apiKey}&units=metric";
```

```
            Console.WriteLine($"[{city}] Requesting weather... (Attempt {attempt})");

            var response = await _httpClient.GetAsync(url);

            if (response.IsSuccessStatusCode)
            {
                var json = await response.Content.ReadAsStringAsync();
                using var doc = JsonDocument.Parse(json);
                var temp =
doc.RootElement.GetProperty("main").GetProperty("temp").GetDecimal();
                var weather =
doc.RootElement.GetProperty("weather")[0].GetProperty("description").GetString();

                return new WeatherData { City = city, Weather = $"{temp}°C, {weather}" };
            }
            else
            {
                throw new HttpRequestException($"API returned {response.StatusCode}");
            }
        }
        catch (Exception ex)
        {
            Console.WriteLine($"[{city}] Attempt {attempt} failed: {ex.Message}");
            if (attempt == retries) break;
            await Task.Delay(delay);
            delay *= 2; // Exponential backoff
        }
    }

    return new WeatherData { City = city, Weather = "Error: Failed after retries" };
  }
  finally
  {
    _semaphore.Release();
  }
 }
}
```

Output :

cities.csv :

cities.csv

```
1  Chennai
2  Bangalore
3  Hyderabad
4  Mumbai
5  Pune
6
```

results.json :

{} results.json > {} 0 > Weather

```
1  [
2    {
3      "City": "Chennai",
4      "Weather": "32.29\u00B0C, broken clouds"
5    },
6    {
7      "City": "Bangalore",
8      "Weather": "34.39\u00B0C, overcast clouds"
9    },
10   {
11     "City": "Hyderabad",
12     "Weather": "38.96\u00B0C, broken clouds"
13   },
14   {
15     "City": "Mumbai",
16     "Weather": "30.37\u00B0C, clear sky"
17   },
18   {
19     "City": "Pune",
20     "Weather": "40.04\u00B0C, scattered clouds"
21   }
22 ]
```