

```

import csv

# Initialize variables to track the lowest and highest life expectancy
min_life_exp = {'Entity': '', 'Year': 0, 'Life Expectancy': float('inf')}
max_life_exp = {'Entity': '', 'Year': 0, 'Life Expectancy': float('-inf')}
data_by_entity = {}

# Open the file and read all lines
with open('life_expectancy.csv', 'r') as file:
    reader = csv.DictReader(file)

    # Process each line in the file
    for row in reader:
        try:
            country = row['Entity']
            year = int(row['Year'])
            life_expectancy = float(row['Life expectancy (years)'])
        except ValueError:
            continue # Skip rows with invalid data

        # Update the lowest and highest life expectancy records
        if life_expectancy < min_life_exp['Life Expectancy']:
            min_life_exp = {'Entity': country, 'Year': year, 'Life Expectancy': life_expectancy}
        if life_expectancy > max_life_exp['Life Expectancy']:
            max_life_exp = {'Entity': country, 'Year': year, 'Life Expectancy': life_expectancy}

        # Organize data by country and year
        if country not in data_by_entity:
            data_by_entity[country] = []
        data_by_entity[country].append({'Year': year, 'Life Expectancy': life_expectancy})

# Initialize variables to track the largest drop
largest_drop = {'Entity': '', 'Year': 0, 'Drop': 0}

# Analyze each country's data to find the largest drop
for country, records in data_by_entity.items():
    records.sort(key=lambda x: x['Year']) # Sort by year
    for i in range(1, len(records)):
        drop = records[i - 1]['Life Expectancy'] - records[i]['Life Expectancy']
        if drop > largest_drop['Drop']:
            largest_drop = {
                'Entity': country,
                'Year': records[i]['Year'],
                'Drop': drop
            }

# Display the lowest and highest life expectancy records
print(f"Lowest Life Expectancy: {min_life_exp['Entity']} in {min_life_exp['Year']} with {min_life_exp['Life Expectancy']} years")
print(f"Highest Life Expectancy: {max_life_exp['Entity']} in {max_life_exp['Year']} with {max_life_exp['Life Expectancy']} years")

# Display the largest drop in life expectancy
print(f"Largest Drop: {largest_drop['Entity']} from {largest_drop['Year']} with a drop of {largest_drop['Drop']} years")

# Allow user to input a year and analyze the data for that year
user_year = int(input("Enter a year to analyze: "))
data_by_year = {}

# Organize data by year
for country, records in data_by_entity.items():
    for record in records:
        year = record['Year']
        if year not in data_by_year:
            data_by_year[year] = []
        data_by_year[year].append({'Entity': country, 'Life Expectancy': record['Life Expectancy']})

if user_year in data_by_year:
    year_data = data_by_year[user_year]
    avg_life_exp = sum(entry['Life Expectancy'] for entry in year_data) / len(year_data)
    min_country = min(year_data, key=lambda x: x['Life Expectancy'])
    max_country = max(year_data, key=lambda x: x['Life Expectancy'])

    print(f"For the year {user_year}:")
    print(f"The average life expectancy across all countries was {avg_life_exp:.3f}")
    print(f"The max life expectancy was in {max_country['Entity']} with {max_country['Life Expectancy']}")
    print(f"The min life expectancy was in {min_country['Entity']} with {min_country['Life Expectancy']}")
else:
    print(f"No data available for the year {user_year}.")

```