T2.1 Complete the attached example task in English with sufficient accuracy for yourself. In the pdf document, tell me freely what it means, i.e. what needs to be done?

Understanding the Assignment

In this task, we are asked to create an application that helps analyze Bitcoin's market data over a given date range. The data will be retrieved from the CoinGecko API. The goal is to provide key insights such as the highest and lowest prices, the highest and lowest trading volume, the longest bullish and bearish trends and the best days for buying and selling Bitcoin.

What Needs to be Done:

Analyze Bitcoin's price data for a specified date range

- We are required to input a start date and an end date.
- The application should process the Bitcoin market data within this date range.

Expected Outputs:

Highest Price and Lowest Price:

- The application should find the dates with the highest and lowest Bitcoin prices within the given date range and display those dates and their corresponding prices.

Highest and Lowest Trading Volume:

- The app must also find and show the dates with the highest and lowest trading volumes for Bitcoin during that period.

Bullish and Bearish Trends:

- The application should identify the longest periods where the Bitcoin price is either consistently increasing (bullish trend) or consistently decreasing (bearish trend). The number of days for each trend should be calculated.

Best Buy and Sell Days:

- The app should suggest the best days to buy and sell Bitcoin to maximize profits. This involves identifying the lowest price for buying and the highest price for selling within the given date range (and vice versa for selling first).

How the Application Should Work:

Input:

- The user will enter a start date and an end date.

Output:

The application will provide the following:

- The date with the highest price and the price on that day.
- The date with the lowest price and the price on that day.
- The date with the highest trading volume and the volume on that day.
- The date with the lowest trading volume and the volume on that day.
- The longest bullish and bearish trends.
- The best days to buy and sell to maximize profits.

The user will interact with the application through input fields or API parameters to get results based on the date range they provide.

Key Points:

- The application must be user-friendly and handle the data accurately or include clear directions (e.g. in a README file) for running the solution.
- The application should be easy to use (whether via a web interface, API, or other tools).
- The code should be clean and simple, minimizing dependencies and focusing on the core functionality.
- The output should be clear and provide Scrooge McDuck with the necessary information to make data-driven decisions about Bitcoin.

T2.2

a) With the help of the task you translated above, explain the meaning of the values of the bitcoin search from and to fields of the coinecko interface below.

The given URL is a request to the CoinGecko API to fetch Bitcoin market data (price, market cap, and volume) in Euros for a specified date range.

Bitcoin search URL:

https://api.coingecko.com/api/v3/coins/bitcoin/market_chart/range?vs_currency=eur&from=1577836800&to=1677836800

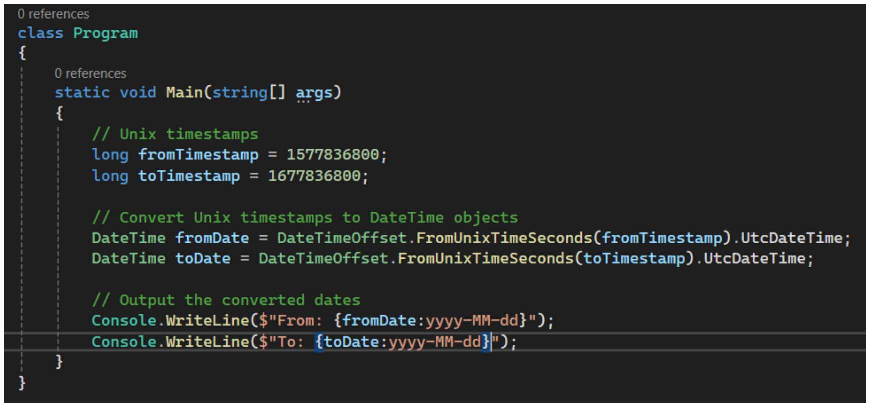
vs_currency=eur:
This specifies that the market data should be returned in Euros (EUR).

from=1577836800:
The from field represents the starting date for the data request. The value 1577836800 is a Unix timestamp.

to=1677836800:
The to field represents the end date for the data request. The value 1677836800 is also a Unix timestamp.

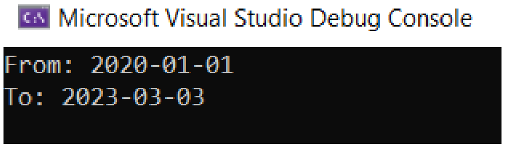
Thus, this URL requests Bitcoin market data (price, volume, and market cap) between January 1, 2020, and March 3, 2023 in Euros.

Example timestamps in C#:



```
0 references
class Program
{
    0 references
    static void Main(string[] args)
    {
        // Unix timestamps
        long fromTimestamp = 1577836800;
        long toTimestamp = 1677836800;

        // Convert Unix timestamps to DateTime objects
        DateTime fromDate = DateTimeOffset.FromUnixTimeSeconds(fromTimestamp).UtcDateTime;
        DateTime toDate = DateTimeOffset.FromUnixTimeSeconds(toTimestamp).UtcDateTime;

        // Output the converted dates
        Console.WriteLine($"From: {fromDate:yyyy-MM-dd}");
        Console.WriteLine($"To: {toDate:yyyy-MM-dd}");
    }
}
```

Output:



```
From: 2020-01-01
To: 2023-03-03
```

b) Adjust the from and to values of the bitcoin search below so that they now refer to this and the previous two weeks.

To adjust the "from" and "to" values so that they refer to this week and the previous two weeks, we need to calculate the Unix timestamps for those dates.

Determine the current date:

- Let’s say today is November 28, 2024 (we can use the current date on our system when making this request).

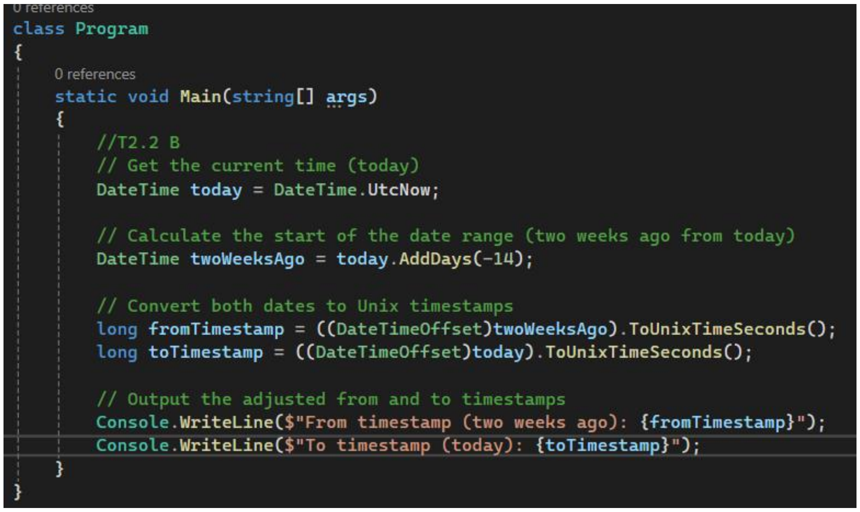
Calculate the Unix timestamp for "today":

- The timestamp for today is the current date and time (in seconds since Unix epoch).

Adjust for "this week and the previous two weeks":

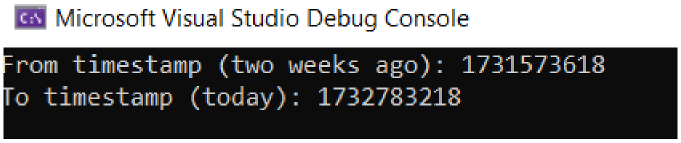
- The start date for "this and the previous two weeks" should be two weeks ago from today.
- The end date would be today.

Example calculate the timestamps in C#:



```
0 references
class Program
{
    0 references
    static void Main(string[] args)
    {
        //T2.2 B
        // Get the current time (today)
        DateTime today = DateTime.UtcNow;

        // Calculate the start of the date range (two weeks ago from today)
        DateTime twoWeeksAgo = today.AddDays(-14);

        // Convert both dates to Unix timestamps
        long fromTimestamp = ((DateTimeOffset)twoWeeksAgo).ToUnixTimeSeconds();
        long toTimestamp = ((DateTimeOffset)today).ToUnixTimeSeconds();

        // Output the adjusted from and to timestamps
        Console.WriteLine($"From timestamp (two weeks ago): {fromTimestamp}");
        Console.WriteLine($"To timestamp (today): {toTimestamp}");
    }
}
```

Ouput:



```
From timestamp (two weeks ago): 1731573618
To timestamp (today): 1732783218
```

The URL for the adjusted request would look like this:

https://api.coingecko.com/api/v3/coins/bitcoin/market_chart/range?vs_currency=eur&from=1731574120&to=1732783720

c) How do you handle their associated values in c#?

In C#, Unix timestamps are handled by using the DateTime and DateTimeOffset classes. The DateTime class is used to represent date and time, and DateTimeOffset is used to ensure that we get the correct Unix timestamp considering time zones.

Here's a step-by-step explanation of how to handle the "from" and "to" timestamps in C#:

Get the Current Date and Time in UTC:

- Use DateTime.UtcNow to get the current date and time in UTC format. This ensures that we are calculating the timestamps based on a consistent time zone (UTC).

Calculate Dates for the Date Range:

- Use AddDays(-7) on the current date to get the date for one week ago.
- The current date (DateTime.UtcNow) will be the "to" value, and the calculated date for one week ago will be the "from" value.

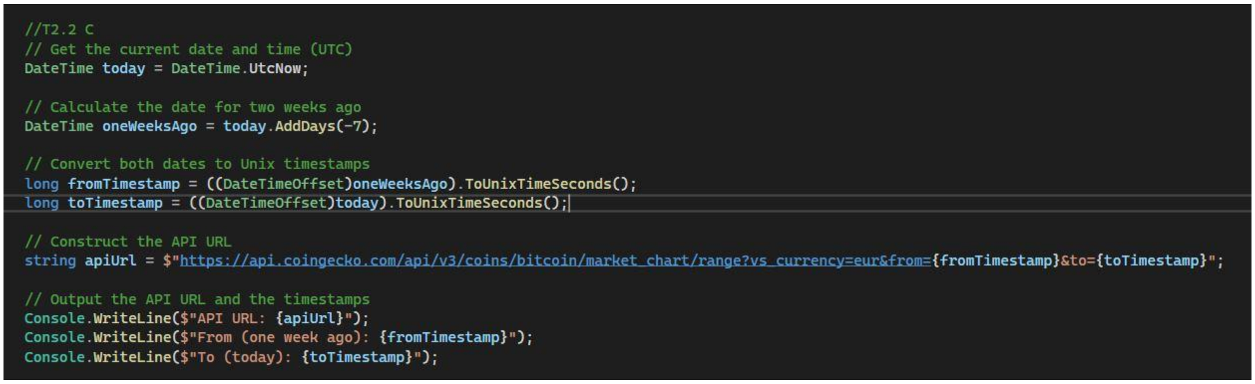
Convert Dates to Unix Timestamps:

- Use DateTimeOffset.ToUnixTimeSeconds() to convert the DateTime values into Unix timestamps.

Form the API URL:

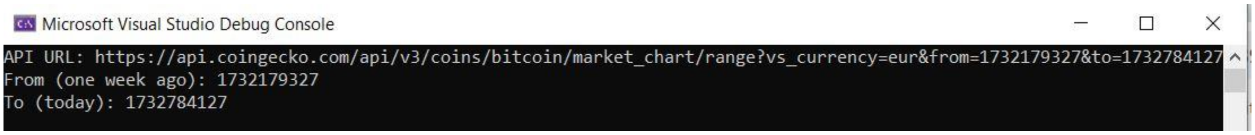
- Construct the URL dynamically by inserting the from and to Unix timestamps into the API request URL.

Example Code in C#:



```
//T2.2 C
// Get the current date and time (UTC)
DateTime today = DateTime.UtcNow;

// Calculate the date for two weeks ago
DateTime oneWeeksAgo = today.AddDays(-7);

// Convert both dates to Unix timestamps
long fromTimestamp = ((DateTimeOffset)oneWeeksAgo).ToUnixTimeSeconds();
long toTimestamp = ((DateTimeOffset)today).ToUnixTimeSeconds();

// Construct the API URL
string apiUrl = $"https://api.coingecko.com/api/v3/coins/bitcoin/market_chart/range?vs_currency=eur&from={fromTimestamp}&to={toTimestamp}";

// Output the API URL and the timestamps
Console.WriteLine($"API URL: {apiUrl}");
Console.WriteLine($"From (one week ago): {fromTimestamp}");
Console.WriteLine($"To (today): {toTimestamp}");
```

Output:



```
API URL: https://api.coingecko.com/api/v3/coins/bitcoin/market_chart/range?vs_currency=eur&from=1732179327&to=1732784127
From (one week ago): 1732179327
To (today): 1732784127
```

The URL for the adjusted request:

https://api.coingecko.com/api/v3/coins/bitcoin/market_chart/range?vs_currency=eur&from=1732179327&to=1732784127