Project Title: Real-Time World Clock Utility in C

Objective:

Develop a **real-time World Clock Utility in C** that continuously displays the **current time, date, day, month, and year** for multiple countries by dynamically adjusting based on **UTC (Coordinated Universal Time) offsets**. The program retrieves UTC time, applies the necessary offset for each country, and updates the display every second while handling potential **day transitions (Next Day or Previous Day).**

Tracked Countries and Time Zones:

The program will display the **local date and time** for the following countries:

Country	**UTC Offset**	**Time Zone Name**
China	UTC+8	CST - China Standard Time
India	UTC+5:30	IST - Indian Standard Time
United Kingdom	UTC+0 / UTC+1 (DST)	GMT/BST - British Summer Time
Argentina	UTC-3	ART – Argentina Time
Saudi Arabia	UTC+3	AST - Arabia Standard Time

Project Requirements:

☑ **Retrieve the current UTC time** from the system clock.
☑ **Apply accurate UTC offsets** for each country (including hours and minutes).
☑ **Handle Day transitions** when time exceeds 24 hours or moves backward.
☑ **Display the full date (day, month, year) along with time**.
☑ **Continuously update time in a loop every second**.
☑ **Ensure formatted console output for readability**.

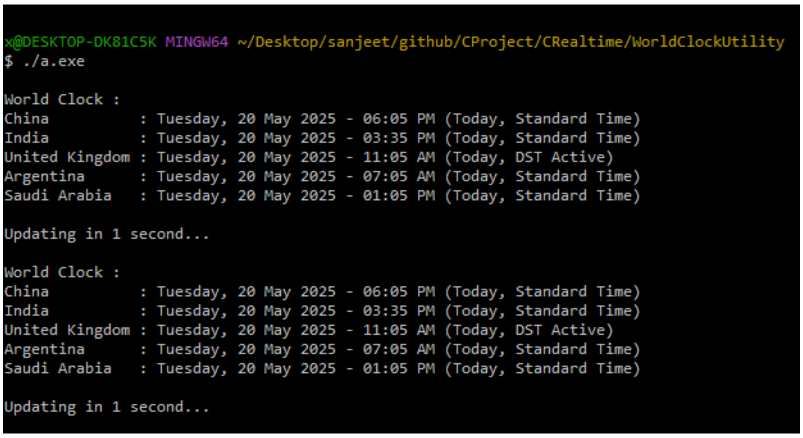
Key Features:

1. **Real-Time Updates** – The clock refreshes every second dynamically.
2. **Multi-Zone Display** – Shows the **local date, day, month, year, and time** for predefined countries.
3. **Day Transition Handling** – Indicates **Next Day or Previous Day** when applicable.
4. **Daylight Saving Time (DST) Consideration** – Adjusts for **regions observing DST shifts** dynamically.
5. **Formatted Console Output** – Displays **structured and readable date/time information**.

C Programming Concepts Used:

◈ **Structures (struct)** → Used for storing **country details**, including UTC offsets, DST settings, and names.
◈ **Arrays** → Used for **managing multiple time zones**, days of the week, and months.
◈ **Time Handling (time.h)** → Retrieves and **processes UTC-based time dynamically**.
◈ **Loops (while)** → Ensures **continuous real-time updates** without manual intervention.
◈ **String Handling (strcpy)** → Manages country names and AM/PM formatting.
◈ **Pointers (*struct)** → Optimizes **memory usage and efficient data handling**.

Expected Output Example:



```
x@DESKTOP-DK81C5K MINGW64 ~/Desktop/sanjeet/github/CProject/CRealtime/WorldClockUtility
$ ./a.exe

World Clock :
China          : Tuesday, 20 May 2025 - 06:05 PM (Today, Standard Time)
India          : Tuesday, 20 May 2025 - 03:35 PM (Today, Standard Time)
United Kingdom : Tuesday, 20 May 2025 - 11:05 AM (Today, DST Active)
Argentina      : Tuesday, 20 May 2025 - 07:05 AM (Today, Standard Time)
Saudi Arabia   : Tuesday, 20 May 2025 - 01:05 PM (Today, Standard Time)

Updating in 1 second...

World Clock :
China          : Tuesday, 20 May 2025 - 06:05 PM (Today, Standard Time)
India          : Tuesday, 20 May 2025 - 03:35 PM (Today, Standard Time)
United Kingdom : Tuesday, 20 May 2025 - 11:05 AM (Today, DST Active)
Argentina      : Tuesday, 20 May 2025 - 07:05 AM (Today, Standard Time)
Saudi Arabia   : Tuesday, 20 May 2025 - 01:05 PM (Today, Standard Time)

Updating in 1 second...
```

Understanding UTC (Coordinated Universal Time):

☑ **What is UTC?**

- UTC (**Coordinated Universal Time**) is the **global standard for timekeeping**, serving as a **fixed reference point**.
- It is **not affected** by **Daylight Saving Time (DST)** changes, unlike regional time zones.
- Local time zones are determined based on **offsets from UTC** (e.g., India follows UTC+5:30).

☑ **How UTC is Used in the Project:**

- The program **retrieves the current UTC time dynamically** using the system clock.
- It **calculates the local date and time** by **adding or subtracting** the appropriate UTC offset.
- Countries like the **United Kingdom** shift their clocks forward during DST, which the program adjusts automatically.
- The utility **handles overflow conditions**, ensuring correct **day/month transitions**.
- The **console output updates every second**, maintaining real-time synchronization.

Future Enhancements:

💡 **User-Defined Time Zones** – Allow users to input **custom UTC offsets** for dynamic tracking.
💡 **Graphical User Interface (GUI)** – Convert **console output into a visually appealing clock**.
💡 **Event Logging** – Save **historical time updates** for **record-keeping**.
💡 **Advanced DST Handling** – Fully automate **seasonal time shifts** based on country-specific rules.

Author & Contact Information:

🧑 **Author: Sanjeet Prasad**
📧 **Email:** sanjeet8.23@gmail.com
📱 **Mobile:** +91 9958217807