Mini Project

**Rainbow Simulation**

1. **Introduction**

This project presents a computer graphics (CG) simulation of a rainbow pattern, showcasing the visual beauty of a rainbow’s colors. Using basic principles of color layering and arc generation, the project aims to provide a visual understanding of rainbow formation through simulation. This project was created using C++ and the graphics.h library.

1. **Objectives**

* Simulate the appearance of a rainbow with distinct color bands.
* Use a series of concentric arcs to represent each color in the rainbow.
* Demonstrate color blending and smooth transitions between colors.
* Provide an animated effect by gradually rendering the arcs in sequence.

1. **Design Approach**

The simulation is developed using the graphics.h library, enabling the creation of 2D graphics in a structured, layered manner to represent the colors of the rainbow.

* Color Layering: Each color in the rainbow (like red, orange, yellow, green, blue, indigo, violet) is represented by an arc. The arcs are layered on top of each other, starting from the largest arc for the outermost color and moving inward for each subsequent color.
* Arc Animation: Each arc is rendered with a slight delay, giving the effect of the rainbow being drawn gradually, enhancing the visual appeal.

1. **Tools and Software**

* Programming Language: C++
* Graphics Library: graphics.h
* Hardware: Standard PC with a basic graphics display capability

1. **Simulation Details**

**5.1 Color Representation**

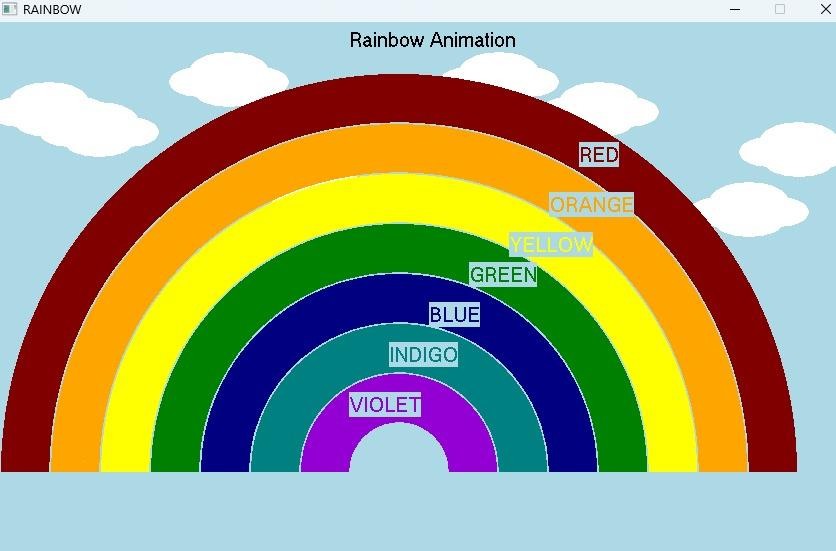
Each arc in the rainbow is represented by a different color using the setcolor() function in graphics.h. Colors include light magenta, light cyan, light gray, light green, light blue, yellow, and light red, to give a vivid representation of the rainbow.

**5.2 Arc Generation**

The arcs are drawn using the arc() function, with each arc having a specific radius to ensure smooth layering.The position and radius of each arc are calculated to align each color closely with the previous one, creating the continuous, curved rainbow effect.

* 1. **Motion Control**

A delay(10) function is used between rendering each arc, adding an animated effect to the rainbow formation as it builds up from the outermost to the innermost arc.

**Output:**

1. **Results and Analysis**

The simulation successfully demonstrates the visual formation of a rainbow with each color represented by a smooth arc.

Each color band is clearly visible, allowing viewers to appreciate the layering and vibrancy typical of a rainbow.

The animation effect adds to the aesthetic, making the rainbow appear gradually, which can engage viewers by simulating a natural phenomenon.

1. **Conclusion**

This project successfully simulates a rainbow pattern using computer graphics. It provides a visual and educational tool for understanding the structure and colors of a rainbow through programming and graphics simulation.