**Experiment Date:** 26-05-2025

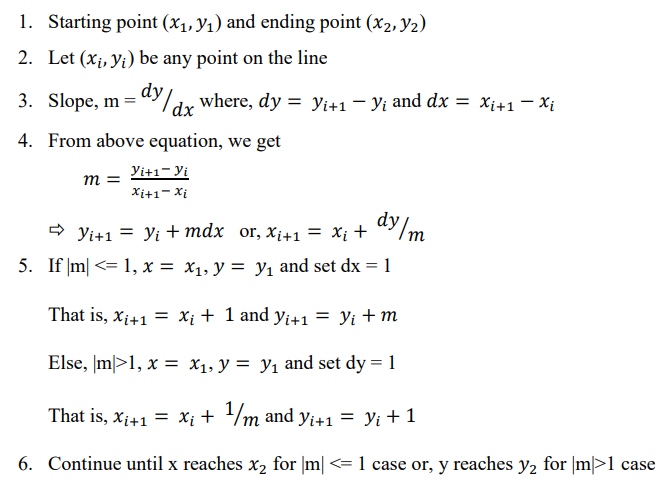
**Experiment Name:** Draw Square using DDA Line Drawing Algorithm

**Description:**

The Digital Differential Analyzer (DDA) algorithm is a simple and accurate method to draw a line between two points using floating-point arithmetic.

To draw a square, we connect its four corners using four DDA-generated lines.

**Algorithm:**

****

**Implementation:**

**A screen shot of a computer program

AI-generated content may be incorrect.**

**A screen shot of a computer program

AI-generated content may be incorrect.**

**Output: A computer screen shot of a computer screen

AI-generated content may be incorrect.**

**Discussion:** In this experiment, we implemented the DDA line drawing algorithm to draw a square. The algorithm divides the line into small steps and calculates intermediate points using floating-point arithmetic, ensuring smooth lines. This method is simple and works effectively for shapes made of straight lines, like squares or rectangles. However, it can be computationally more expensive due to floating-point operations compared to Bresenham's algorithm.

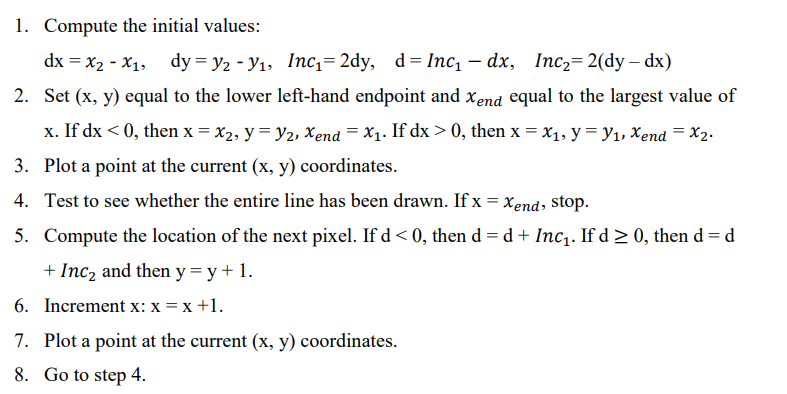
**Experiment Date:** 16-06-2025

**Experiment Name:** Draw rectangle Bresenham line drawing algorithm

**Description:** Bresenham’s line algorithm is an *integer* incremental scan‑conversion method. It decides the next pixel using only additions/subtractions and comparisons—no floating points—so it’s faster than DDA.

To draw a rectangle, we just draw 4 Bresenham lines: bottom, right, top, and left.

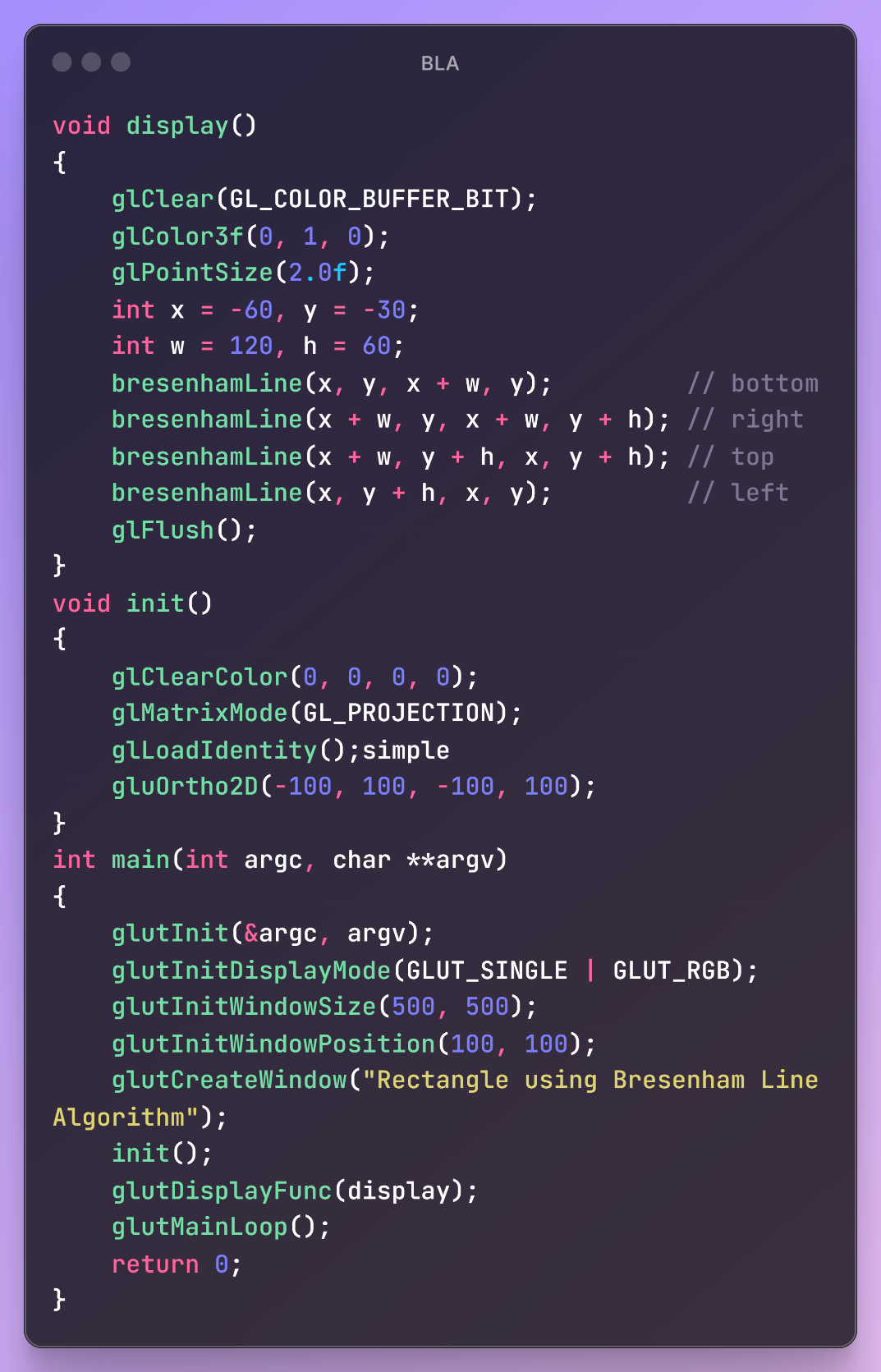
**Algorithm:**

****

**Implementation:**

**A screenshot of a computer program

AI-generated content may be incorrect.**

****

**Output:**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Discussion:**

In this experiment, we used Bresenham's line drawing algorithm to draw a rectangle. The algorithm uses only integer arithmetic to determine the nearest pixel to approximate the line, which makes it faster and more efficient than DDA. It is highly suitable for systems with limited processing power and works accurately for shapes like rectangles composed of straight edges.

**Experiment Date:** 23-06-2025

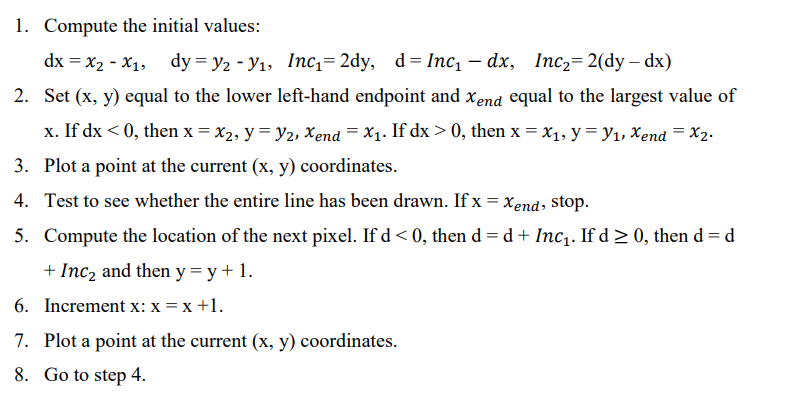
**Experiment Name:** Draw national flag using Bresenham line drawing and Bresenham circle drawing algorithm

**Description:**

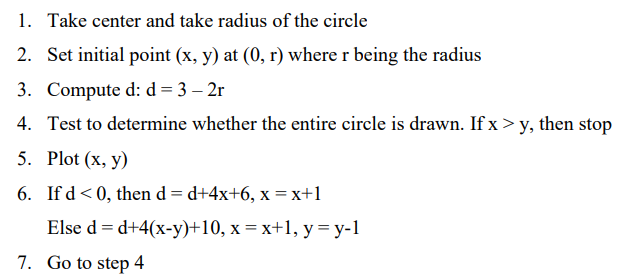
To draw the national flag of Bangladesh, we combine two efficient rasterization techniques — Bresenham’s Line Drawing Algorithm for the flag’s rectangular body and Bresenham’s Circle Drawing Algorithm for the red circle at the center-left.

These algorithms use only integer operations, which makes them optimal for digital displays and embedded systems where floating-point calculations are costly.

**Algorithm:**

**Bresenham Line Drawing:** 

**Bresenham Circle Drawing:**



**Implementation:**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**Output:**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Discussion:**In this experiment, we drew the national flag of Bangladesh by combining two Bresenham algorithms — one for lines and one for circles. The rectangle part of the flag was drawn using Bresenham’s line algorithm, while the red circle was rendered using the circle drawing algorithm. This showed how multiple algorithms can be used together to construct complex figures with high precision and minimal computational cost.

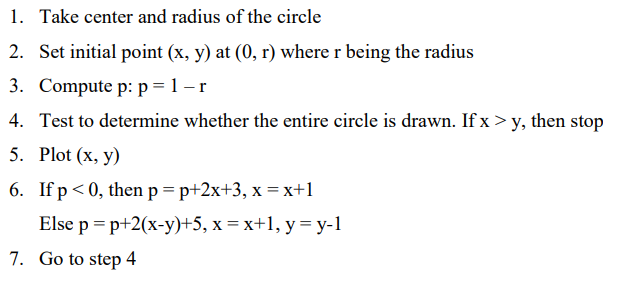
**Experiment Date:**23-06-2025

**Experiment Name:** Mid-point circle drawing algorithm

**Description:**

The Midpoint Circle Algorithm is a rasterization technique that uses decision parameters and symmetry to efficiently plot points on a circle using only integer arithmetic. This method avoids floating-point operations and trigonometric functions, making it suitable for low-resource graphics systems.

**Algorithm:**



**Implementation:**

A screenshot of a computer program

AI-generated content may be incorrect.

**Output:**

A screenshot of a computer

AI-generated content may be incorrect.

**Discussion:**

In this experiment, we implemented the Midpoint Circle Drawing Algorithm to draw a circle. The algorithm uses a decision parameter to select the next pixel, leveraging symmetry to reduce computations. Unlike floating-point-based methods, it only uses integer arithmetic, making it faster and well-suited for real-time applications where circles are frequently used, such as in GUIs and game graphics.