

## EXERCISE - 1

**AIM :** To implement a line-drawing algorithm (e.g., DDA or Bresenham's algorithm) in C to draw a straight line between two points.

### Procedure (Using Bresenham's Algorithm)

1. Input:
  - o Two endpoints (x1,y1) and (x2,y2).
2. Calculate the differences:
  - o  $\Delta x = x_2 - x_1$
  - o  $\Delta y = y_2 - y_1$
3. Determine the decision parameters:
  - o Set  $p = 2\Delta y - \Delta x$  for the initial decision variable.
4. Iteratively plot points:
  - o Start from the first endpoint and move toward the second endpoint.
  - o Based on the decision variable  $p$ , determine whether to increment the y-coordinate.
5. Repeat until the second endpoint is reached.

### SAMPLE CODE:

```
#include <stdio.h>
#include <graphics.h>

void bresenhamLine(int x1, int y1, int x2, int y2) {
    int dx = x2 - x1;
    int dy = y2 - y1;

    int p = 2 * dy - dx; // Initial decision parameter
    int x = x1, y = y1;

    // Plot the first point
    putpixel(x, y, WHITE);

    // Iterate through the points
    while (x < x2) {
        x++;
```

```

        if (p < 0) {
            p += 2 * dy; // Mid-point below the line
        } else {
            y++;
            p += 2 * (dy - dx); // Mid-point above or on the line
        }
        putpixel(x, y, WHITE); // Plot the next point
    }
}

int main() {
    int gd = DETECT, gm;
    int x1, y1, x2, y2;

    // Initialize the graphics system
    initgraph(&gd, &gm, "C:\\TURBOC3\\BGI");

    // Input endpoints
    printf("Enter the coordinates of the first point (x1, y1): ");
    scanf("%d %d", &x1, &y1);
    printf("Enter the coordinates of the second point (x2, y2): ");
    scanf("%d %d", &x2, &y2);

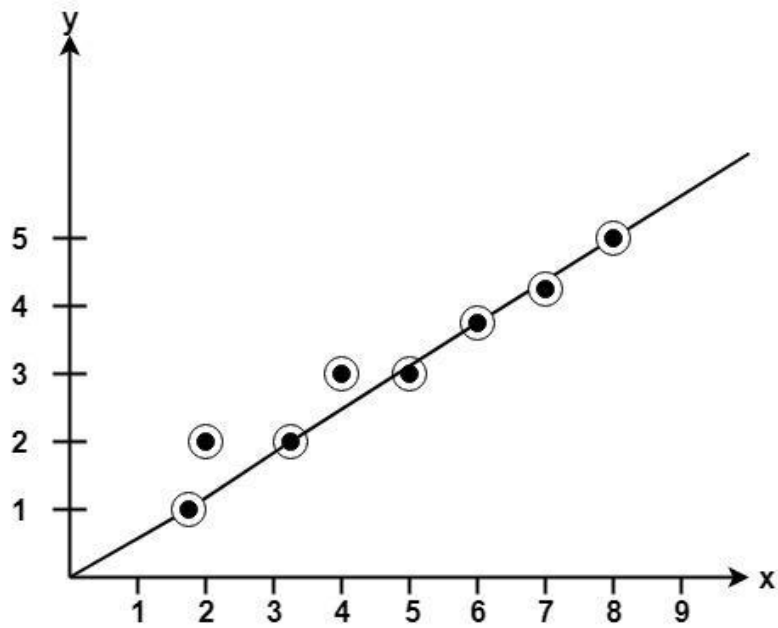
    // Draw the line using Bresenham's algorithm
    bresenhamLine(x1, y1, x2, y2);

    // Wait for user input to close the graphics window
    getch();
    closegraph();

    return 0;
}

```

## OUTPUT



```
NeuTroN DOS-C++ 0.77, Cpu speed: max 100% cycles, Frameskip 0, Program:
Enter co-ordinates of first point: 100
100
Enter co-ordinates of second point: 200
200
```