



## Experiment 2

**Student Name:** Gaganjot Singh

**UID:** 22BCS14843

**Branch:** B.E. CSE III Yr

**Section:** 22BCS-IOT-612-B

**Semester:** 6<sup>th</sup>

**Subject Name:** Computer Graphics with Lab

**Subject Code:** 22CSH-352

- 1. Aim:** Implement and compare the performance of Simple DDA, Symmetrical DDA, and Bresenham's algorithm for positive and negative line slope.
- 2. Objective:** The objective of this practical is to implement and compare the performance of Simple DDA, Symmetrical DDA, and Bresenham's line-drawing algorithms for lines with both positive and negative slopes. The comparison focuses on computational efficiency, accuracy, and their ability to render lines on a raster display.

### **3. Code:**

```
#include <iostream.h>
#include <conio.h>
#include <graphics.h>
#include <math.h>

void simpleDDA(int x1, int y1, int x2, int y2)
{
    int dx = x2 - x1;
    int dy = y2 - y1;
    int steps = (abs(dx) > abs(dy)) ? abs(dx) : abs(dy);
    float xInc = dx / (float)steps;
    float yInc = dy / (float)steps;
    float x = x1, y = y1;
    for (int i = 0; i <= steps; i++)
        {putpixel((int)(x + 0.5), (int)(y + 0.5), WHITE);
    x += xInc; y += yInc;}}

void symmetricalDDA(int x1, int y1, int x2, int y2)
```



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
{ int dx = x2 - x1;
    int dy = y2 - y1;
    int steps = (abs(dx) > abs(dy)) ? abs(dx) : abs(dy);
    float xInc = dx / (float)steps;
    float yInc = dy / (float)steps;
    float x = x1, y = y1;

    for (int i = 0; i <= steps; i++)
    {
        putpixel((int)(x + 0.5), (int)(y + 0.5), YELLOW);
        x += xInc;
        y += yInc;
    }

void bresenham(int x1, int y1, int x2, int y2)
{
    int dx = abs(x2 - x1);
    int dy = abs(y2 - y1);
    int sx = x1 < x2 ? 1 : -1;
    int sy = y1 < y2 ? 1 : -1;
    int err = dx - dy;

    while (1)
    { putpixel(x1, y1, GREEN);
    if (x1 == x2 && y1 == y2)
        break;
    int e2 = 2 * err;
    if (e2 > -dy){
        err -= dy;
        x1 += sx;}
    if (e2 < dx)
        {err += dx;
        y1 += sy;}}}
```



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Discover. Learn. Empower.

```
void main()
{   int c;
    clrscr();
    int gd = DETECT, gm;
    initgraph(&gd, &gm, "C:\\\\TurboC3\\\\BGI");

    int x1, y1, x2, y2;
    outtextxy(150, 450, "Name: Gaganjot Singh | Roll No: 22BCS14843");

    // Input points for the line
    cout << "Enter the starting point (x1, y1): ";
    cin >> x1 >> y1;
    cout << "Enter the ending point (x2, y2): ";
    cin >> x2 >> y2;

do {
    cout << "Choose the line drawing algorithm:\n";
    cout << "1. Simple DDA\n";
    cout << "2. Symmetrical DDA\n";
    cout << "3. Bresenham's Algorithm\n";
    cout << "4. Exit\n"; // Option to exit
    cout << "Enter your choice (1-4): ";
    cin >> c;

    switch (c) {
        case 1:
            simpleDDA(x1, y1, x2, y2);
            break;
        case 2:
            symmetricalDDA(x1, y1, x2, y2);
            break;
        case 3:
```

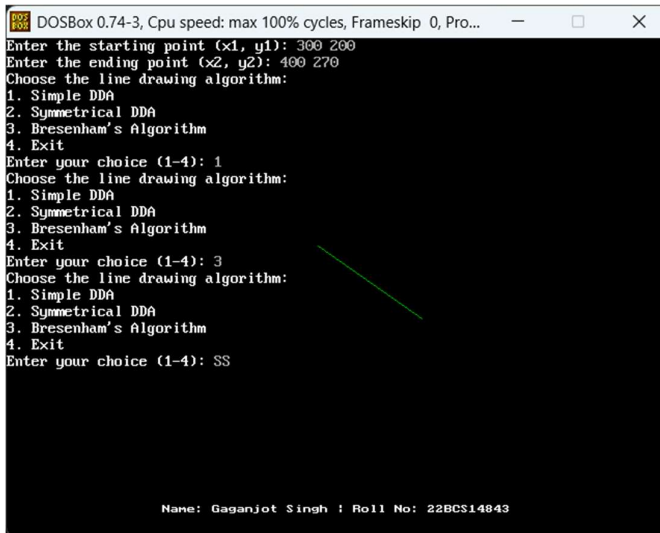
```

        bresenham(x1, y1, x2, y2);
        break;
    case 4:
        cout << "Exiting the program." << endl;
        break;
    default:
        cout << "Invalid choice! Please select a valid option (1-4)."
<<endl;

        break;}
} while (c != 4); // Continue until the user chooses to exit
getch();
closegraph();}

```

## 4. Output:



```

DOSBox 0.74-3, Cpu speed: max 100% cycles, Frameskip 0, Pro...
Enter the starting point (x1, y1): 300 200
Enter the ending point (x2, y2): 400 270
Choose the line drawing algorithm:
1. Simple DDA
2. Symmetrical DDA
3. Bresenham's Algorithm
4. Exit
Enter your choice (1-4): 1
Choose the line drawing algorithm:
1. Simple DDA
2. Symmetrical DDA
3. Bresenham's Algorithm
4. Exit
Enter your choice (1-4): 3
Choose the line drawing algorithm:
1. Simple DDA
2. Symmetrical DDA
3. Bresenham's Algorithm
4. Exit
Enter your choice (1-4): 4
Name: Gaganjot Singh : Roll No: 22BCS14843

```

## 5. Learning Outcome:

- Analyze and compare the efficiency of Simple DDA, Symmetrical DDA, and Bresenham's algorithm.
- Use of Switch case statements for make a menu driven code.
- Gain proficiency in drawing fundamental shapes such as lines, circles, rectangles, arcs, and ellipses using the graphics.h library.
- Developed the ability to precisely position these shapes within the graphics window.
- Use of Putpixel for putting pixel along with colors.