Title: To use Bresenham's Line Algorithm to draw a line between given points.

Theory:

A line segment between two points (x1, y1) and (x2, y2) in a 2D plane is traditionally a continuous connection. However, in computer graphics, pixels are discrete units, so we cannot directly draw a perfect continuous line between these points. Instead, we approximate the line by determining and drawing the closest pixels that best represent the line path.

In graphics programming, the output screen acts as a coordinate system with the origin (0, 0) at the top-left corner. The x-coordinate increases to the right, and the y-coordinate increases downward.

Bresenham's Line Algorithm (BLA) is an efficient method to compute the intermediate pixel coordinates that form a straight line between two points. It uses integer calculations to minimize computational overhead and ensures smooth, visually appealing lines by deciding the optimal pixels to activate.

Using functions like `putpixel(x, y, color)` in C, BLA allows us to render line segments by illuminating the appropriate pixels on the screen, producing a precise and performance-friendly line drawing.

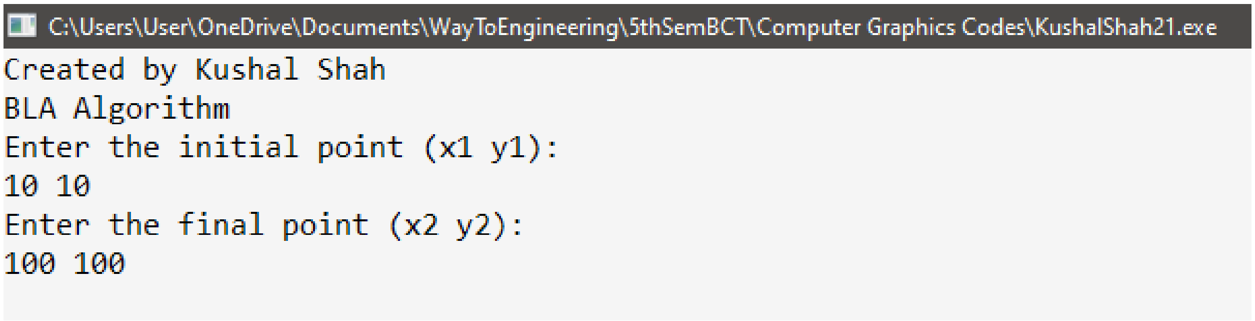
Bresenham's Line Algorithm:

```
Step 1:  Start
Step 2:  Take the coordinates of initial point (x1,y1) and the final point (x2,y2).
Step 3:  Find the difference between the points by using the formula:
                dx = x2-x1;
                dy = y2-y1;
Step 4:  Calculate the decision parameter:
         p = 2*dy-dx;
Step 5:  Repeat the following for i=0 to dx:
          putpixel(x1,y1,K);
          x1 = x1+1;
          If (p<0)
                p=p+2*dy;
          Else
                y=y+1;
                p=p+2*(dy-dx);
Step 6:  Stop
```

Source Code:

```
// Bresenham's Line Algorithm to draw a line.
#include <stdio.h>
#include <graphics.h>
#include <math.h>

int main(int argc, char const *argv[])
{
   int x1, y1, x2, y2, dx, dy, p, x, y;
   printf("Created by Kushal Shah\nBLA Algorithm\n");

   printf("Enter the initial point (x1 y1):\n");
   scanf("%d %d", &x1, &y1);

   printf("Enter the final point (x2 y2):\n");
   scanf("%d %d", &x2, &y2);

   int gd = DETECT, gm;
   initgraph(&gd, &gm, NULL);

   dx = x2 - x1;
   dy = y2 - y1;

   x = x1;
   y = y1;

   p = 2 * dy - dx;

   outtextxy(10, 110, "Kushal Shah");

   for (int i = 0; i <= dx; i++)
   {
      putpixel(x, y, WHITE);
      delay(100);
      x++;
      if (p < 0)
      {
         p = p + 2 * dy;
      }
      else
      {
            y++;
            p = p + 2 * (dy - dx);
        }
    }

    delay(500);
    closegraph();
    return 0;
}
```

Output:



C:\Users\User\OneDrive\Documents\WayToEngineering\5thSemBCT\Computer Graphics Codes\KushalShah21.exe

```
Created by Kushal Shah
BLA Algorithm
Enter the initial point (x1 y1):
10 10
Enter the final point (x2 y2):
100 100
```

Figure 1: Inserting initial and final points

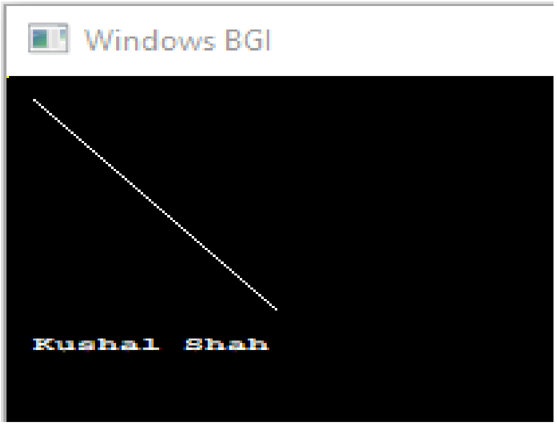


Figure 2: Drawing line using BLA

Conclusion:

Thus, as shown in the program above, we can draw a line by plotting individual pixels using Bresenham's Line Algorithm (BLA) with the graphics functions provided in the graphics.h header file.