Name - Kritika Gaur.

Course - BCA '6' - C

Roll no - 17 (1121075)

Subject - Comp. Graphics.

Paper Code - TBC602

Date - 16/06/2021

Q-1 Bresenham's Line Algorithm.

Step 1: Start algorithm.

Step 2: Declare variable $x_1, x_2, y_1, y_2, d, i_1, i_2$, dx, dy.

Step 3: Enter value of x_1, y_1, x_2, y_2.

where x_1, y_1 are coordinates of starting point and x_2, y_2 are coordinate of ending point

Step 4: Consider (x, y) as starting point and x_{end} as maximum possible value of x.

If $dx < 0$
Then $x = x_2$.
$y = y_2$
$x_{end} = x_1$.

If $dx > 0$
Then $x = x_1$.
$y = y_1$.
$x_{end} = x_2$.

Step 5: Generate point at (x, y) coordinates

Step 6: Check if whole line is generated.

If $x >= x_{end}$
stop.

Step 7: Calculate co-ordinates of the next pixel

If $d > 0$
Then $d = d + i_1$.
increment $y = y + 1$.

If $d \geq 0$
Then $d = d + i_2$.

Step 8: Increment $x = x + 1$.

Step 9: Draw a point of latest (x, y) coordinates

Step 10: Go to step 7.

Step 11: End of algorithm.

Program:

```
#include <stdio.h>
#include <graphics.h>
void drawline (int x0, int y0, int x1, int y1)
{
int dx, dy, p, x, y;
dx= x1-x0;
dy= y1-y0;
 x=x0;
 y=y0;
p= 2 * dy -dx;
while ( x<x1)
{
if (p>=0).
{ putpixel (x, y, 7);
 y=y+1;
 p = p+2*dy - 2 *dx;
}
else
```

```
{
    putpixel (x, y, 7);
    p = p + 2 * dy;
    x = x + 1;
  }
}

int main()
{
int gdriver = DETECT, gmode, error, x0, y0, x1, y1;
initgraph (&gdriver, &gmode, "c:\\turboc3\\bgi");
printf("Enter co-ordinate of first point:");
scanf("%d %d", &x0, &y0);
printf("Enter co-ordinates of second point:");
scanf("%d %d", &x1, &y1);
drawline (x0, y0, x1, y1);
return 0;
}
```