

Name: Vineet Pal

Roll No: 1121167

Course: BCA VI 'C'

## Ans 1) Bresenham's Line Algorithm

Step 1) Start

Step 2) Declare variable  $x_1, x_2, y_1, y_2, dx, dy, ix, ia, dx, dy$

Step 3) Enter value of  $x_1, y_1, x_2, y_2$

( $x_1, y_1$  are starting point co-ordinates)

( $x_2, y_2$  are ending point co-ordinates)

Step 4) Calculate  $dx = x_2 - x_1$

$$dy = y_2 - y_1$$

$$ia = dx \div dy$$

$$ir = dx \div (dy - dx)$$

$$j = i - dx$$

Step 5) Consider ( $x_1, y_1$ ) as starting point &  $x_{end}$  as minimum possible value of  $x$

if  $dx < 0$

then  $x = x_2$

$$y = y_2 \quad x_{end} = x_1$$

if  $dx > 0$

then  $x = x_1$

$$y = y_1 \quad x_{end} = x_2$$

Step 6 > Generate point at  $(n, y)$  co-ordinates

Step 7 > Check if whole line is generated

If  $n \neq n_{end}$

Stop.

Step 8 > Calculate co-ordinates of next pixel

If  $\delta < 0$

then  $\delta = \delta + i_1$

If  $\delta > 0$

then  $\delta = \delta + i_2$

increment  $y = y + 1$

Step 9 > increment  $n = n + 1$

Step 10 > Draw a point of latest  $(n, y)$  coordinates

Step 11 > Go to step 7

Step 12 > End

Ans

```
#include <stdio.h>
#include <graphics.h>
void drawline(int x0, yo int y0, int x1, int y1)
{
    int dx, dy, p, m, y;
    dx = x1 - x0;    x = x0;
    dy = y1 - y0;    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);

```

$P = P + dx \cdot dy;$

$n = n + 1;$

} }

int main()

{

int gdriver = DETECT, gmode, error, x0, y0, x1, y1;

initgraph(&gdriver, &gmode, "c:\\turbo\\c31\\bgi");

printf("Enter co-ordinates 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;

}

