

Name:- Mukul Balani
Course:- BCA '6' Sem-2
Roll No:- 1121085 (23)
Subject:- Computer Graphic.
Date:- 16/06/2021
Stodid:- 18211293

Ans. 2 Bresenham's Line Algorithm:

Step 1:- Start Algorithm

Step 2:- Declare variable $x_1, y_1, x_2, y_2, dx, ix, iy, dx, dy$.

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 coordinates of ending point.

Step 4:- Calculate $dx = x_2 - x_1$

Calculate $dy = y_2 - y_1$

Calculate $ix = 2 * dy$

Calculate $iy = 2 * dx$

Calculate $dx = ix - dx$

Calculate $dy = iy - dy$

Step 5:- Consider (x, y) as starting point and x end as maximum possible value of x

if $dx < 0$

then $x = x_2$

$y = y_2$

Mukul

$x_{end} = x_1$

If $d > 0$

Then $x = x_1$

$y = y_1$

$x_{end} = x_2$

Step 6:- Generate Point at (x, y) Coordinates

Step 7:- Check if whole line is generated

if $x \neq x_{end}$

Stop

Step 8:- Calculate Co-ordinates of the next pixel.

If $d < 0$

Then $d = d + i_1$

If $d > 0$

Then $d = d + i_2$

Increment $y = y + 1$

Step 9:- Increment $x = x + 1$

Step 10:- Draw a point of latest (x, y) Coordinates

Step 11:- Go to step 7

Step 12:- End of Algorithm.

Signature

Program

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

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

```
void drawing (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, 1);
```

```
            y = y + 1;
```

```
            p = p + 2 * dy - 2 * dx;
```

```
}
```

```
        else
```

```
{
```

```
            Putpixel (x, y, 1);
```

```
            p = p + 2 * dy;
```

```
}
```

```
            x = x + 1;
```

```
}
```

Signature

