Ans 1 →

Bresenham's Line Algorithm :

Step 1 → Start Algorithm

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

Step 3 → Enter value of x_1, y_1, x_2, y_2

where x_1, y_1 are coordinates of starting point

x_2, y_2 are coordinates of Ending point

Step 4 → Calculate $dx = x_2 - x_1$

Calculate $dy = y_2 - y_1$

Calculate $i_1 = 2 * dy$

Calculate $i_2 = 2 * (dy - dx)$

Calculate $d = i_1 - dx$

Step 5 → Consider (x, y) as starting point and xend as maximum possible value of x.

```
if $dx < 0$
    then $x = x_2$
         $y = y_2$
         $xend = x_1$

if $dx > 0$
    then $x = x_1$
         $y = y_1$
         $xend = x_2$
```

Step 6 → Generate point at (x, y) coordinates.

Step 7 → Check if whole line is generated

```
    if x >= xend
    Stop
```

Step 8 → Calculate co-ordinates of the next pixel

```
    if d < 0
        then d = d + i1
    if d ≥ 0
        then d = d + i2
        increment y = y + 1
```

Step 9 → Increment x = x + 1

Step 10 → Draw a point of latest (x, y) coordinates

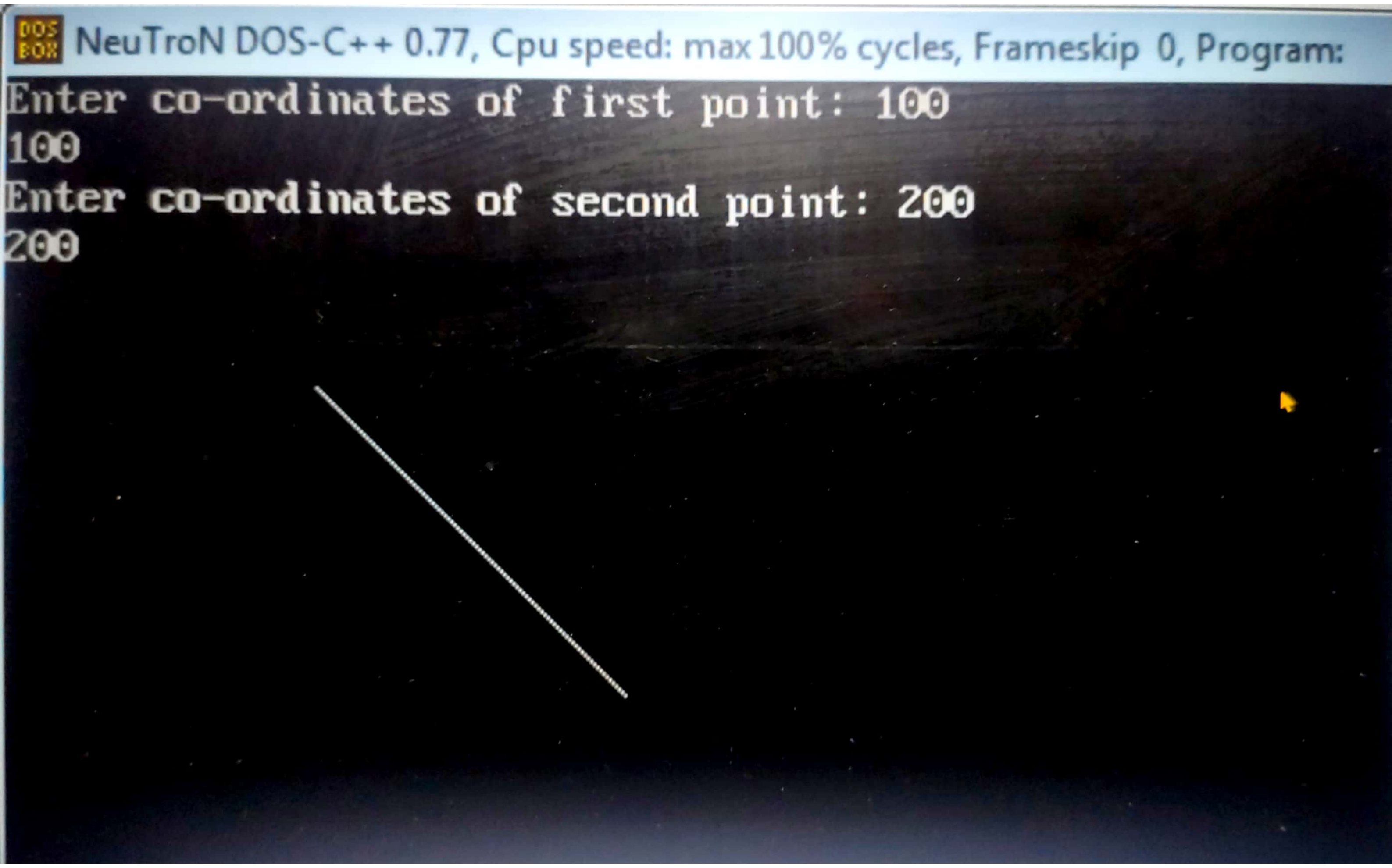
Step 11 → Go to step 7.

Step 12 → End of Algorithm

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 gd = DETECT, gm , x0, y0, x1, y1 ;
    initgraph ( &gd , &gm , " " );
    printf (" Enter Co-ordinates of first point : ");
    scanf ( "%d %d", &x0 , &y0);
    printf ( " Enter coordinates of second point : ");
    scanf( "%d %d", &x1 , &y1 );
    drawline (x0, y0, x1 , y1 );
    return 0;
}
```


NeuTroN DOS-C++ 0.77, Cpu speed: max 100% cycles, Frameskip 0, Program:

```
Enter co-ordinates of first point: 100
100
Enter co-ordinates of second point: 200
200
```