Name :- Aniruddh Rana
Course :- BCA 'A'
Univ. Roll No. :- 1121018 '16'
Subject :- Computer Graphics & Animation (PBC-602)
Sem. :- VI th

Aniruddh

Problem / Solution :- DDA Algorithm

- Starting coordinate = (x_0, y_0)
- Ending coordinate = (x_n, y_n)

The points generation using DDA Algorithm involves the following steps :-

Step-01 :- Calculate Δx, Δy and m from the given input.
We know that the slope of a straight line m is given as :

These parameters are calculated as :

- $\Delta x = x_n - x_0$
- $\Delta y = y_n - y_0$
- $m = \Delta y / \Delta x \Rightarrow m = \frac{y_n - y_0}{x_n - x_0}$

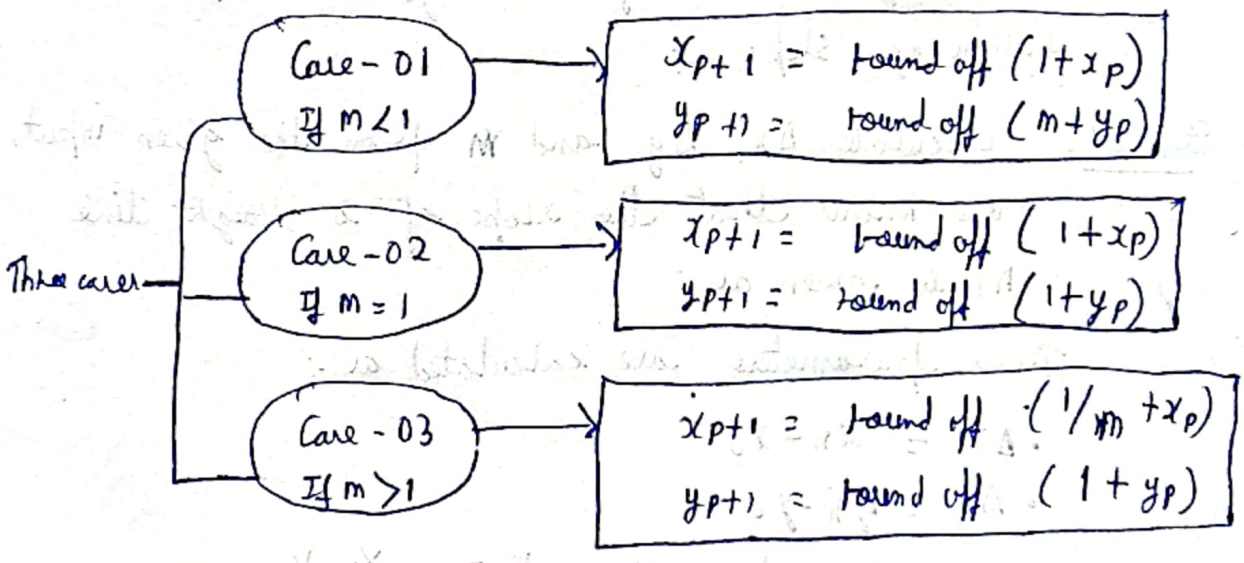
Step-02 :- Find the number of steps or points in between the starting and ending co-ordinates.

```
if (absolute (Δx) > absolute (Δy))
    steps = absolute (Δx);
else
    steps = absolute (Δy);
```

Name -: Aniruddh Rana
Course -: BCA 'A'
Univ. Roll No. -: 1121018 '16'
Subject -: Computer Graphics & Animation (PBC-602)
Sem. -: VI th
Aniruddh

Step-03 : Suppose the current point is (x_p, y_p) and the next point is (x_{p+1}, y_{p+1}).

Find the next by following below three cases;



Step-04 : Keep repeating step-03 until the end point is reached or the number of generated new points (including the starting and ending points) equals to the steps count.

Name :- Aniruddh Rana
Course :- BCA 'A'
Univ. Roll No. :- 1121018 '16'
Subject :- Computer Graphics & Animation (PBC-602)
Sem. :- VI th
Aniruddh

DDA program :

```
#include <stdio.h>
#include <graphics.h>
   int main ()
   { int rou ( float num)
      { return num < 0? num - 0.5 : num + 0.5;
      }
int x1 = 100, x2 = 250, y1 = 100, y2 = 250, step;
int gd = DETECT, gm;
   float x, y, m;
    int dx = x2 - x1;
    int dy = y2 - y1;
     m = dy/dx ;
    if ( dx > dy )
          step = dx;
    else
        step = dy;
initgraph (&gd, &gm, " ");
outtextxy (x1, y1, "A");
```

Name :- Aniruddh Rana
Course :- BCA 'A'
Univ. Roll No. :- 1121018 '16'
Subject :- Computer Graphics & Animation (PBC-602)
Sem. :- VI th
Aniruddh

```
outtextx (x2, y2, "B");
x = x1 , y = y1 ;
while (step > 0)
{   if (m < 1)
    {  x = x + 1;
       y = y + m;
    } if (m >= 1)
{
    x = x+1 / m;
     y = y + 1 ;
}
putpixel (iou(x), iou (y), RED);
 step --;
}
getch ();
return 0;
}
```

OUTPUT