Q1

## DDA Algorithm -

Step1 - Start Algorithm.

Step2 - Declare $x_1, y_1, x_2, y_2, x, y, dx, dy$ as integers.

Step3 - Enter value of $x_1, y_1, x_2, y_2$.

Step4 - Calculate dx

i.e - $dx = x_2 - x_1$

Step5 - Calculate dy

i.e - $dy = y_2 - y_1$

Step6 - If abs (dx) > abs (dy)

then step = abs (dx)

else

Step7 - $x_{inc} = \frac{dx}{step}$

$y_{inc} = \frac{dy}{step}$

assign $x = x_1$

assign $y = y_1$

Step8 - Set pixel (x, y)

Step9 - $x = x + x_{inc}$

$y = y + y_{inc}$

Set pixels (Round (x), Round (y))

Step 10 - Repeat step 9 until $x = x_2$

Step 11 . Stop.

## Program

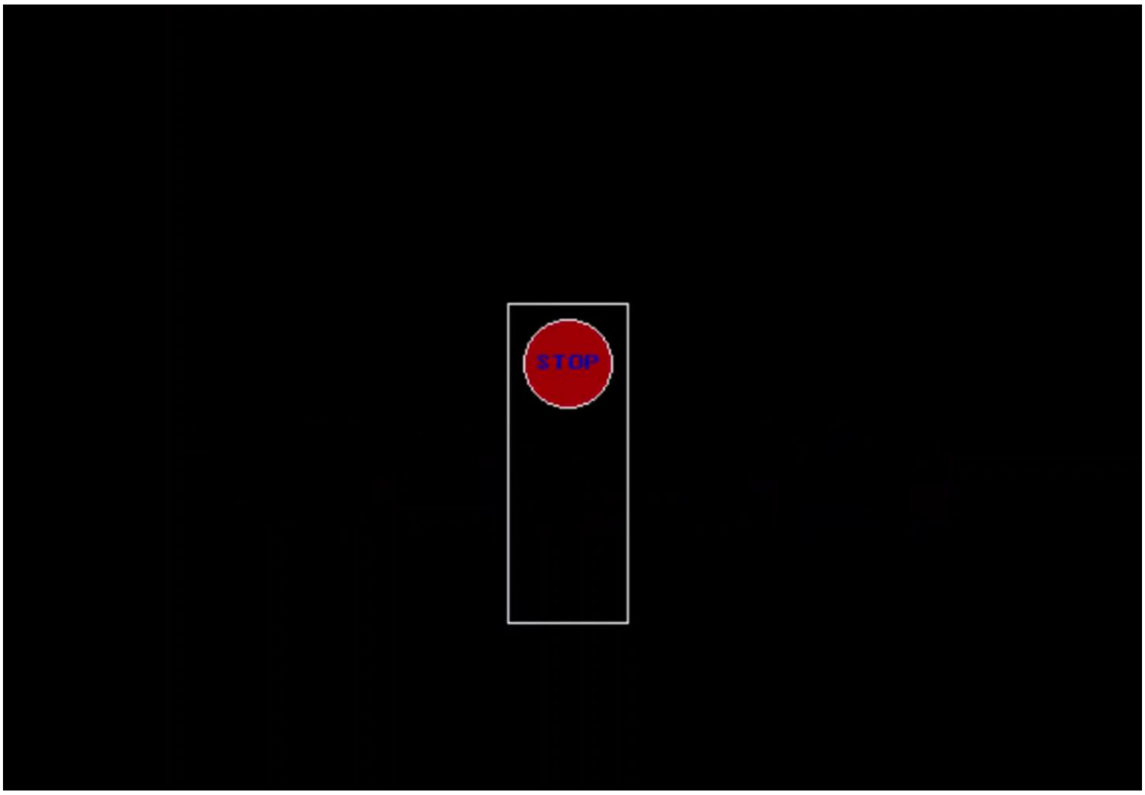
```
# include <stdio.h>
# include <graphics.h>
int main ()
{ int gd = DETECT, gm;
  float x, y, dx, dy, steps;
  int x0, y0, x1, y1;
  initgraph ($gd, $gm, "C:\TURBOC3\BGI");
   x0 = 100, y0 = 200;
   x1 = 500, y1 = 300;
   dx = (float)(x1 - x0);
   dy = (float)(y1 - y0);
   if (dx >= dy)
    steps = dx;
   else
    steps = dy;
    dx = dx / steps;
    dy = dy / steps;
    x = x0;
    y = y0;
int i = 1;
while (i ≤ steps)
 {
   putpixel (x, y, PURPLE);
   x = x + dx;
   y = y + dy;
   i = i + 1;
 }
   getch ();
   close graph ();
}
```

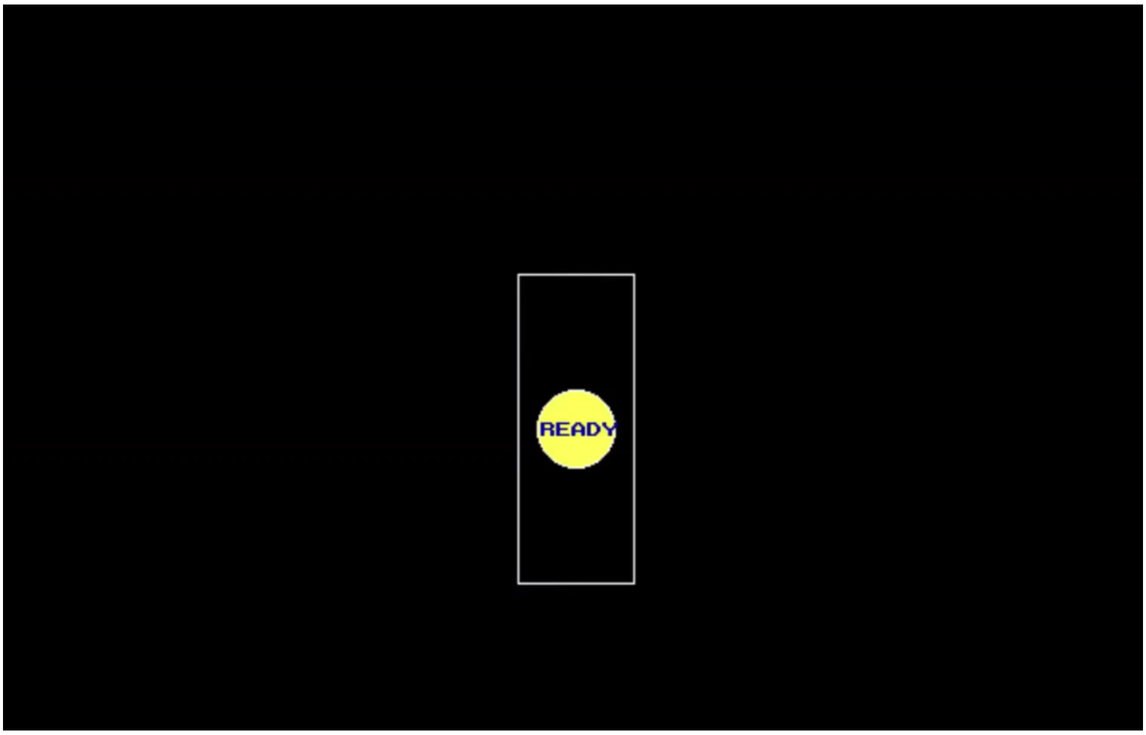
## Q3 Traffic light with animation -

```
#include <stdio.h>
#include <graphics.h>
int main()
{ int gd= DETECT, gm ,midx, midy;
  initgraph (&gd , &gm, "C:\TURBOC3\BGI");
  midx=getmaxx()/2;
  midy= getmaxy ()/2;
  Setcolor (WHITE);
  rectangle (midx -30, midy - 80 ,midx +30,
                  midy +80;
  circle (midx , midy-50 , 22);
  Setfill style (SOLID_FILL , RED);
  flood fill (midx , midy-50, 22);
  Setcolor (BLUE);
  Outtextxy (midx, midy -50 , "STOP");
  delay (2000);
  clear device ();
  Setcolor (WHITE);
  rectangle (midx -30, midy - 80, midx +30,
                  midy + 80);
Circle (midx, midy-50, 22);
Setfillstyle (SOLID-Fill, Yellow);
floodfill (midx, midy, WHITE);
 Setcolor (BLUE);
Outtextxy (midx-18, midy-3, "READY");
 delay (2000);
 Clear device ();

 Set color (WHITE);
 rectangle (midx-30, midy-80, midx+30,
              midy+80);
  Circle (midx, midy+50, 22);
  Setfillstyle (SOLID-Fill, GREEN);
  floodfill (midx, midy+50, WHITE);
   Setcolor (BLUE);
  Outtextxy (midx-7, midy+48, "GO");

   getch();
   close graph();
}
```


STOP


READY