

Name: Geetanjali Bisht  
University Roll no: 1221054  
Paper Code: PBC-602

Course: BCA 'A'  
Subject: Computer Graphics & Animation

END TERM Practical

Ans 1) Algorithm: To implement DDA Line Drawing Algorithm.

Step 1: Declare  $x_1, y_1, x_2, y_2, dx, dy, x, y$  as integer variables.

Step 2: Enter the values of  $x_1, y_1, x_2, y_2$ .

Step 3: Calculate  $dx = x_2 - x_1$

Step 4: Calculate  $dy = y_2 - y_1$

Step 5: If  $ABS(dx) > ABS(dy)$   
Then  $step = dx$   
Else

Step 6:  $x_{inc} = dx / step$   
 $y_{inc} = dy / step$

assign  $x = x_1$

assign  $y = y_1$

Step 7: set pixel  $(x, y)$

Step 8 :  $x = x + x_{inc}$

$y = y + y_{inc}$

set pixels (Round (x), Round (y))

Step 9 : Repeat step 9 until  $x = x_2$

Step 10 : End Algorithm.

Program :

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

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

```
int main()
```

```
int round (float a)
```

```
{ return (a < 0 ? a - 0.5 : a + 0.5);
```

```
}
```

```
int main()
```

```
{ int gd = DETECT, gm;
```

```
float x1, y1, x2, y2, steps, dx, dy, xinc, yinc;
```

```
printf("Enter the starting & ending co-ordinates of line\n");
```

```
scanf("%f %f %f %f %f %f", &x1, &y1, &x2, &y2);
```

initgraph (&gd, &gm, NULL);

$dx = x_2 - x_1;$

$dy = y_2 - y_1;$

$fabs(dx);$

$fabs(dy);$

if ( $fabs(dx) > fabs(dy)$ )

{  $step = fabs(dx);$

}

{  $steps = fabs(dy);$

}

$nxinc = dx / steps;$

$nyinc = dy / steps;$

putpixel ( $x_1, y_1, Red$ );

for (int  $i = 0; i < steps; i++$ )

{  $x_1 = x_1 + nxinc;$

$y_1 = y_1 + nyinc;$

putpixel (round off ( $x_1$ ), round off ( $y_1$ ),  $i+1/40$ );

delay(50);

}

getch ();

drawgraph ();

return 0;

}

