Name - Himanshu Baghari | Subject Name - Computer Graphics and Animation
U. Roll No - 1121060 | Subject Code - PEC-602

Ans 1. Bresenham line Drawing Algorithm -

→ Algorithm -

Step 1. Start algorithm.

Step 2. Declare ~~value~~ (variable) $x_1, x_2, y_1, y_2, d, i_1, i_2, dx, dy$

Step 3. Enter value of x_1, y_1, x_2, y_2
where x_1, y_1 are coordinates of starting point.
and x_2, y_2 are coordinates of ending point.

~~Step 4.~~

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 x_{end} as maximum possible value of x

if $dx < 0$
Then $x = x_2$
$y = y_2$
$x_{end} = x_1$

if $dx > 0$
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 $x >= x_{end}$

stop.

Step 8. Calculate co-ordinates of the next pixel.

if $d < 0$

Then $d = d + i_1$

if $d \geq 0$

then $d = d + i_2$

Increament $y = y + 1$

Step 9. Increament $x = x + 1$

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

Step 11. Go to step 7

Step 12. End of Algorithm

→ Code :-

```
#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 gdriver = DETECT, gmode, error, x0, y0, x1, y1;
   initgraph (&gdriver, &gmode, "c:\\turboc3\\ bgi");
  printf("Enter co-ordinates of first point:");
  scanf("%d %d", &x0, &y0);
  printf("Enter co-ordinates of second point:");
  scanf (" %d %d. "&x1, &y1);
 drawline (x0, y0, x1, y1);
 return 0;
}
```