

Name - Nitin Rawat
Course - BCA
Section - 'C'
University Roll no - 1121095
Subject - CG Practical

Q01. Ans:-

Algorithm

Step 1:- Start

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

Step 3:- Enter Value of x_1, x_2, y_1, y_2

Where x_1, y_1 Coordinate of starting point
and x_2, y_2 " " end 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
 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) Coordinate

Step 7:- Check if whole line is generated
if $x \geq x_{end}$

Stop
Step 8:- Calculate Co-ordinate of the pixel
if $d < 0 \rightarrow$ then $d = d + 1$ if $d \geq 0$
Then $d = d + 2$

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)
```

```

    { put pixel (x, y, 7)
      y = y + 1;
      p = p + 2 * dy - 2 * dx;

```

```

    }
  else

```

```

    { put pixel (x, y, 7);
      p = p + 2 * dy; }

```

```

      x = x + 1;

```

```

    } }

```

```

int main ( )

```

```

{
  int gdriver = DETECT, gmode, error, x0, y0, x1, y1;
  initgraph (&gdriver, &gmode, "tutoc311bgi");
  printf ("Enter Co-ordinate of first point");
  scanf ("%d %d", &x0, &y0);
  printf ("Enter second point of Co-ordinate");
  scanf ("%d %d", &x1, &y1);
  drawline (x0, y0, x1, y1);
  return 0;
}

```

Name - Nitin Rawat

Course - BCA

Section - 'C'

University Roll no - 1121095

Subject - C++ Practical

Q02 Ans:-

Algorithm:-

$$p_i = f(x_i + 1, y_i - \frac{1}{2}) \\ = (x_i + 1)^2 + (y_i - \frac{1}{2})^2 - r^2 \\ \text{equation 2}$$

Step 1:- put $x=0$, $y=r$ in equation 2
We have $p=1-r$

Step 2:- Repeat steps while $x \leq y$

plot (x, y)

if $(p < 0)$

then set $p = p + 2x + 3$

else

$p = p + 2(y - x) + 5$

$y = y - 1$ (end if)

$x = x + 1$ (end loop)

Step 3:- end

Program:-

```
#include <stdio.h>
void mid point Circle draw (int x_Centre,
int y_Centre, int r)
{
    int x = r, y = 0;
    printf ("%d %d", x + x_Centre, y + y_Centre);
    if (r > 0)
    {
        printf ("%d %d", x + x_Centre, -y + y_Centre);
        printf ("%d %d", y + x_Centre, x + y_Centre);
        printf ("%d %d\n", -y + x_Centre, x + y_Centre);
    }
    int p = 1 - r;
    while (x > y)
    {
        y++;
        if (p <= 0)
            p = p + 2 * y + 1;
        else
        {
            x--;
            p = p + 2 * y - 2 * x + 1;
        }
    }
```



```
if (x < y) {
```

```
break;
```

```
printf("%.1f, %.1f", x + x_Centre, y + y_Centre);
```

```
printf("%.1f, %.1f", -x + x_Centre, y + y_Centre);
```

```
printf("%.1f, %.1f", x + x_Centre, -y + y_Centre);
```

```
printf("%.1f, %.1f\n", -x + x_Centre, -y + y_Centre);
```

```
if (x != y) {
```

```
{
```

```
printf("%.1f, %.1f", y + x_Centre, x + y_Centre);
```

```
printf("%.1f, %.1f", -y + x_Centre, x + y_Centre);
```

```
printf("%.1f, %.1f", y + x_Centre, -x + y_Centre);
```

```
printf("%.1f, %.1f\n", -y + x_Centre, -x + y_Centre);
```

```
}
```

```
}
```

```
}
```

```
int main ()
```

```
{
```

```
Midpoint Circle Draw(0, 0, 3);
```

```
return 0;
```

```
}
```