

Name - Vaibhav Joshi

Rollno - 42 C1121157

Course - BCA

Section - C

Sem - 6

Subject - Computer graphics

Algo - Given coordinate of two points $A(x_1, y_1)$ and $B(x_2, y_2)$. The task is to find all the intermediate points required to draw AB on screen.

```
#include <bits/stdc++.h>
```

```
using namespace std;
```

```
void bresenham (int x1, int y1, int x2, int y2)
```

```
{  
    int m-new = 2 * (y2 - y1);
```

```
    int slope_error_new = m-new - (x2 - x1);
```

```
    for (int x = x1; y = y1; x <= x2; x++)
```

```
    {  
        cout << " (" << x << ", " << y << ") \n";
```

```
        slope_error_new += m-new;
```

```
        if (slope_error_new >= 0)
```

```
        {  
            y++;
```

```
            slope_error_new = 2 * (x2 - x1);
```

```
        }
```

```
    }
```

```
}
```

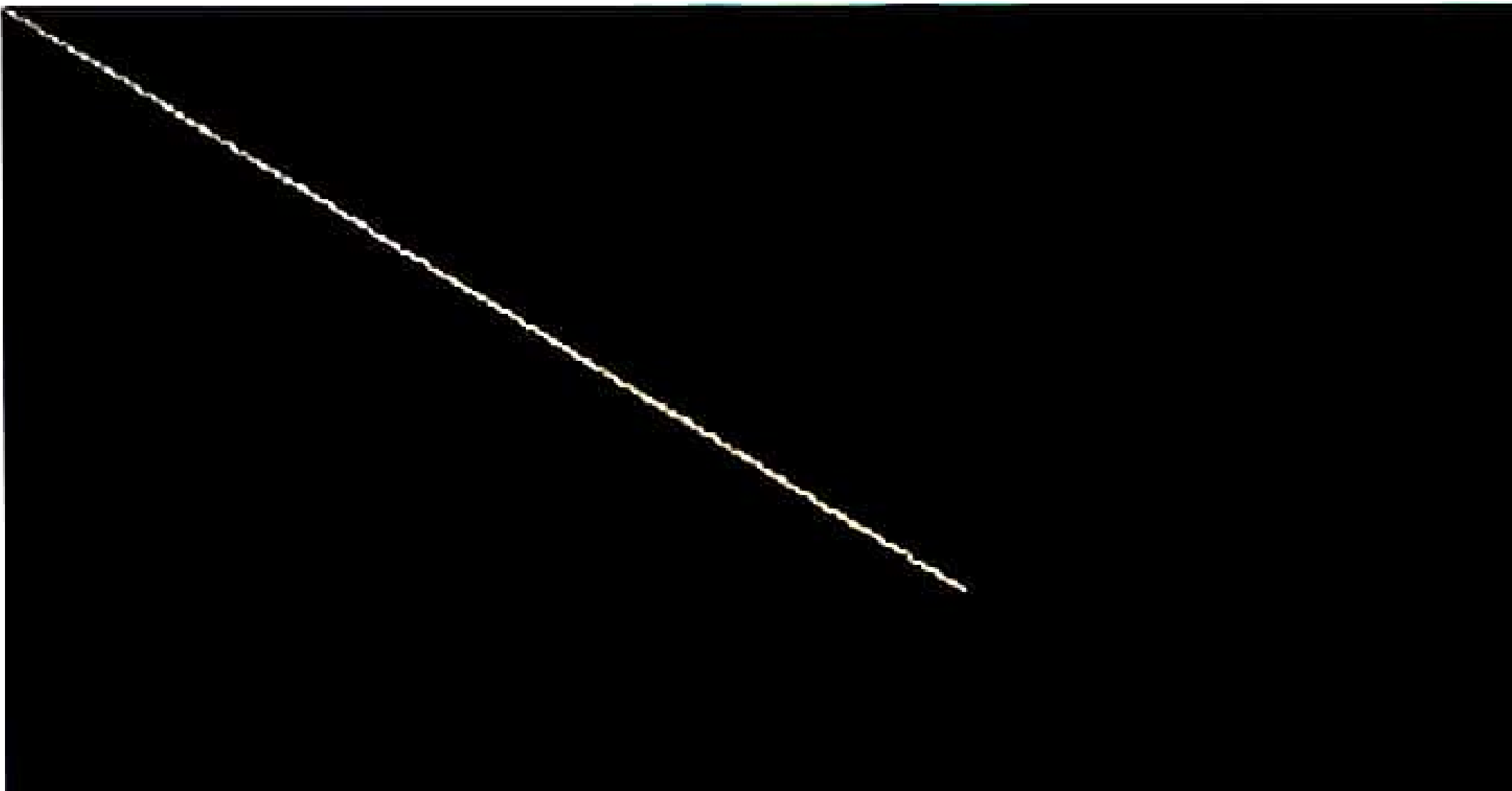
```
int main()
```

```
{  
    int x1 = 3, y1 = 2, x2 = 15, y2 = 5;
```

```
    bresenham (x1, y1, x2, y2);
```

```
    return 0;
```

```
}
```



Name - Vaibhav Joshi

Rollno - 42 C11211571

Course - BCA

Section - C

Sem - 6

Subject - Computer graphics

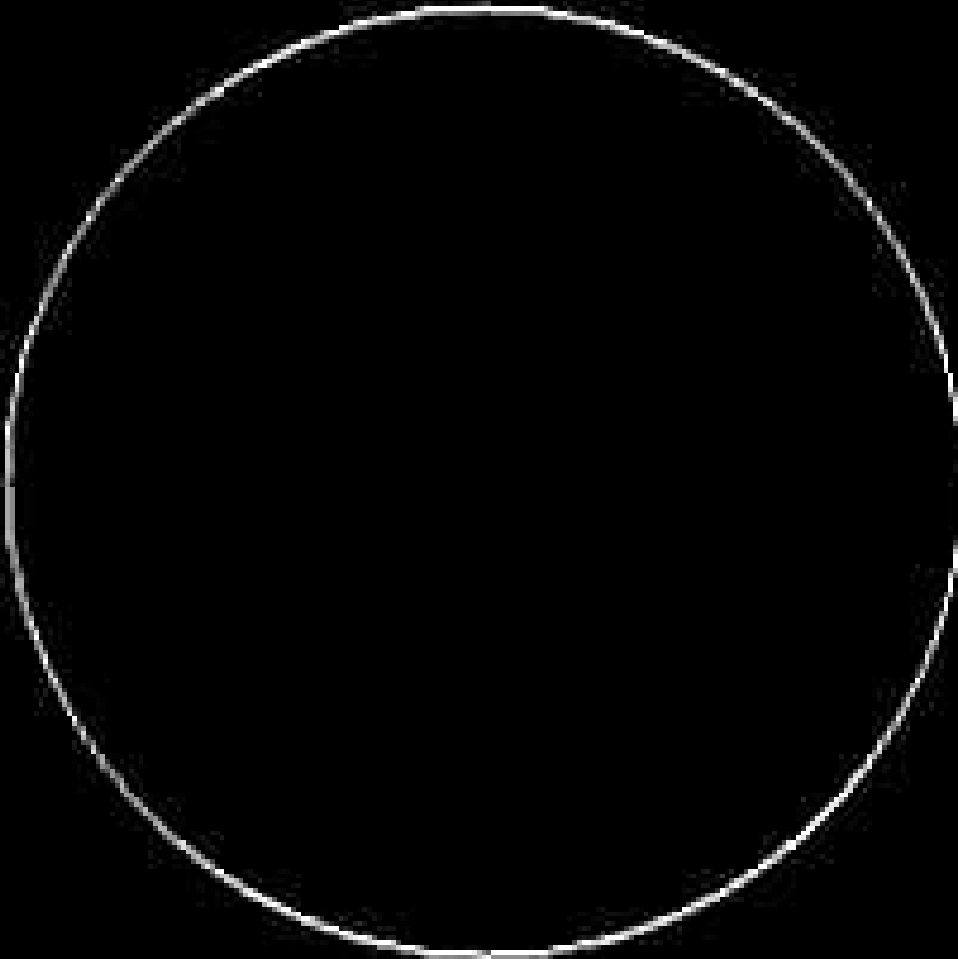
Algo - The mid point circle algorithm is an algorithm used to determine the point needs for rasterizing a circle.

```
#include <stdio.h>
void midPointCircular 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("( (%d, %d)", x+x-centre, y+y-centre);
        printf("( (%d, %d)", -x+x-centre, y+y-centre);
        printf("( (%d, %d)", x+x-centre, -y+y-centre);
        printf("( (%d, %d)\n", -x+x-centre, -y+y-centre);
        if (x!=y)
        {
            printf("( (%d, %d)", y+x-centre, x+y-centre);
            printf("( (%d, %d)", -y+x-centre, x+y-centre);
            printf("( (%d, %d)", y+x-centre, -x+y-centre);
            printf("( (%d, %d)\n", -y+x-centre, -x+y-centre);
        }
    }
}

int main()
{
    int x-centre=0, y-centre=0, r=10;
    midPointCircular draw(x-centre, y-centre, r);
}
```

Enter radius of circle: 100

Enter co-ordinates of center(x and y): 150
150



Name - Vaibhav Joshi

Course - BCA

Sem - 6

Subject - Computer graphics

Rollno - 42 C1121157

Section - C

Algo - Boundary fill algorithm starts at a pixel inside the polygon to be filled and paints the interior proceeding outwards towards the boundary.

```
#include <graphic.h>
void boundaryfill4(int x, int y, int fill_color, int boundary_color)
```

```
{
    if (getpixel(x, y) != boundary_color &&
        getpixel(x, y) != fill_color)
```

```
{
    putpixel(x, y, fill_color);
    boundaryfill4(x+1, y, fill_color, boundary_color);
    boundaryfill4(x, y+1, fill_color, boundary_color);
    boundaryfill4(x-1, y, fill_color, boundary_color);
    boundaryfill4(x, y-1, fill_color, boundary_color);
}
```

```
}
int main()
```

```
{
    int gd = DETECT, gm;
    initgraph(&gd, &gm, "");
    int x = 250, y = 200, radius = 50;
    circle(x, y, radius);
    boundaryfill4(x, y, 6, 15);
    delay(10000);
    getch();
    closegraph();
    return 0;
```

```
}
```

