

Experiment -1: Scan Conversion of Circle using Midpoint algorithm

Code:

```
#include <bits/stdc++.h>
#include<iostream>
#include <graphics.h>
void drawCircle(int xc, int yc, int radius)
{
    int x = 0;
    int y = radius;
    int decision = 1 - radius;
    while (y >= x)
    {
        putpixel(xc + x, yc + y, WHITE);
        putpixel(xc - x, yc + y, WHITE);
        putpixel(xc + x, yc - y, WHITE);
        putpixel(xc - x, yc - y, WHITE);
        putpixel(xc + y, yc + x, WHITE);
        putpixel(xc - y, yc + x, WHITE);
        putpixel(xc + y, yc - x, WHITE);
        putpixel(xc - y, yc - x, WHITE);

        if (decision <= 0)
        {
            x++;
            decision += 2 * x + 1;
        }
        else
        {
            y--;
            x++;
            decision += 2 * (x - y) + 1;
        }
    }
}

int main()
{
    //cout<<"____Mid-Point Circle Algorithm____"<<endl;
    int gd = DETECT, gm;
    initgraph(&gd, &gm, "");
```

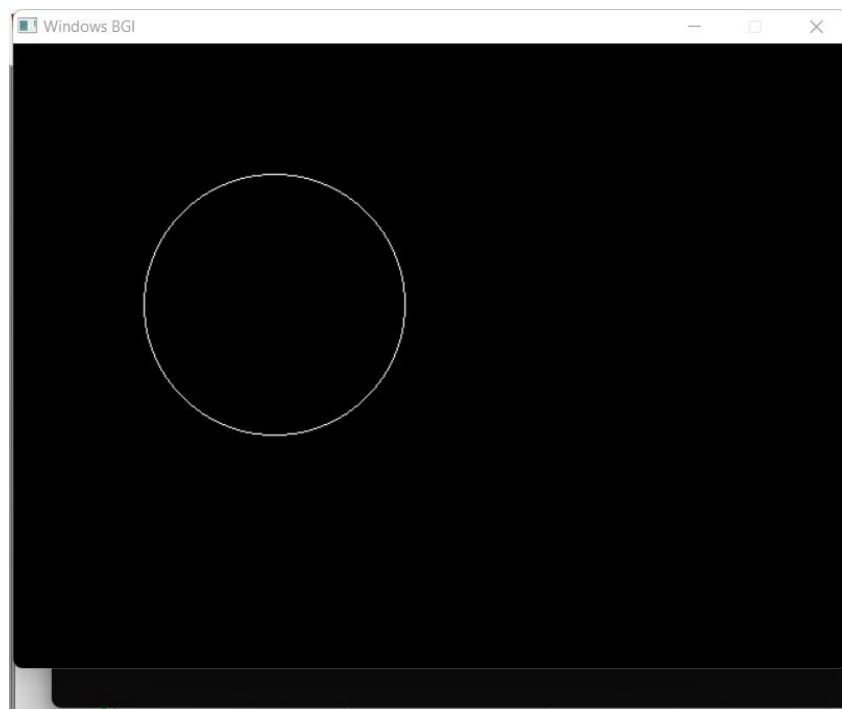
```
int xc, yc, radius;
std::cout << "Enter the center coordinates (xc, yc): ";
std::cin >> xc >> yc;
std::cout << "Enter the radius: ";
std::cin >> radius;

drawCircle(xc, yc, radius);

delay(5000);
closegraph();

return 0;
}
```

Output:



Experiment -2: Scan Conversion of Ellipse using Midpoint algorithm

Code:

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

void drawEllipse(int xc, int yc, int rx, int ry)
{
    int x = 0;
    int y = ry;

    // Decision parameter for region 1
    int d1 = (ry * ry) - (rx * rx * ry) + (0.25 * rx * rx);

    int dx = 2 * ry * ry * x;
    int dy = 2 * rx * rx * y;

    while (dx < dy)
    {
        putpixel(xc + x, yc + y, WHITE);
        putpixel(xc - x, yc + y, WHITE);
        putpixel(xc + x, yc - y, WHITE);
        putpixel(xc - x, yc - y, WHITE);

        if (d1 < 0)
        {
            x++;
            dx = dx + (2 * ry * ry);
            d1 = d1 + dx + (ry * ry);
        }
        else
        {
            x++;
            y--;
            dx = dx + (2 * ry * ry);
            dy = dy - (2 * rx * rx);
            d1 = d1 + dx - dy + (ry * ry);
        }
    }
}
```

```

// Decision parameter for region 2
int d2 = ((ry * ry) * ((x + 0.5) * (x + 0.5))) + ((rx * rx) * ((y - 1) * (y - 1))) - (rx * rx *
ry * ry);

while (y >= 0)
{
    putpixel(xc + x, yc + y, WHITE);
    putpixel(xc - x, yc + y, WHITE);
    putpixel(xc + x, yc - y, WHITE);
    putpixel(xc - x, yc - y, WHITE);

    if (d2 > 0)
    {
        y--;
        dy = dy - (2 * rx * rx);
        d2 = d2 + (rx * rx) - dy;
    }
    else
    {
        y--;
        x++;
        dx = dx + (2 * ry * ry);
        dy = dy - (2 * rx * rx);
        d2 = d2 + dx - dy + (rx * rx);
    }
}
}

```

```

int main()
{
    int gd = DETECT, gm;
    initgraph(&gd, &gm, "");

    int xc, yc, rx, ry;
    std::cout << "Enter the center coordinates (xc, yc): ";
    std::cin >> xc >> yc;
    std::cout << "Enter the x-radius (rx): ";
    std::cin >> rx;
    std::cout << "Enter the y-radius (ry): ";
    std::cin >> ry;

    drawEllipse(xc, yc, rx, ry);

    delay(5000);
}

```

```
    closegraph();  
    return 0;  
}
```

Output:

