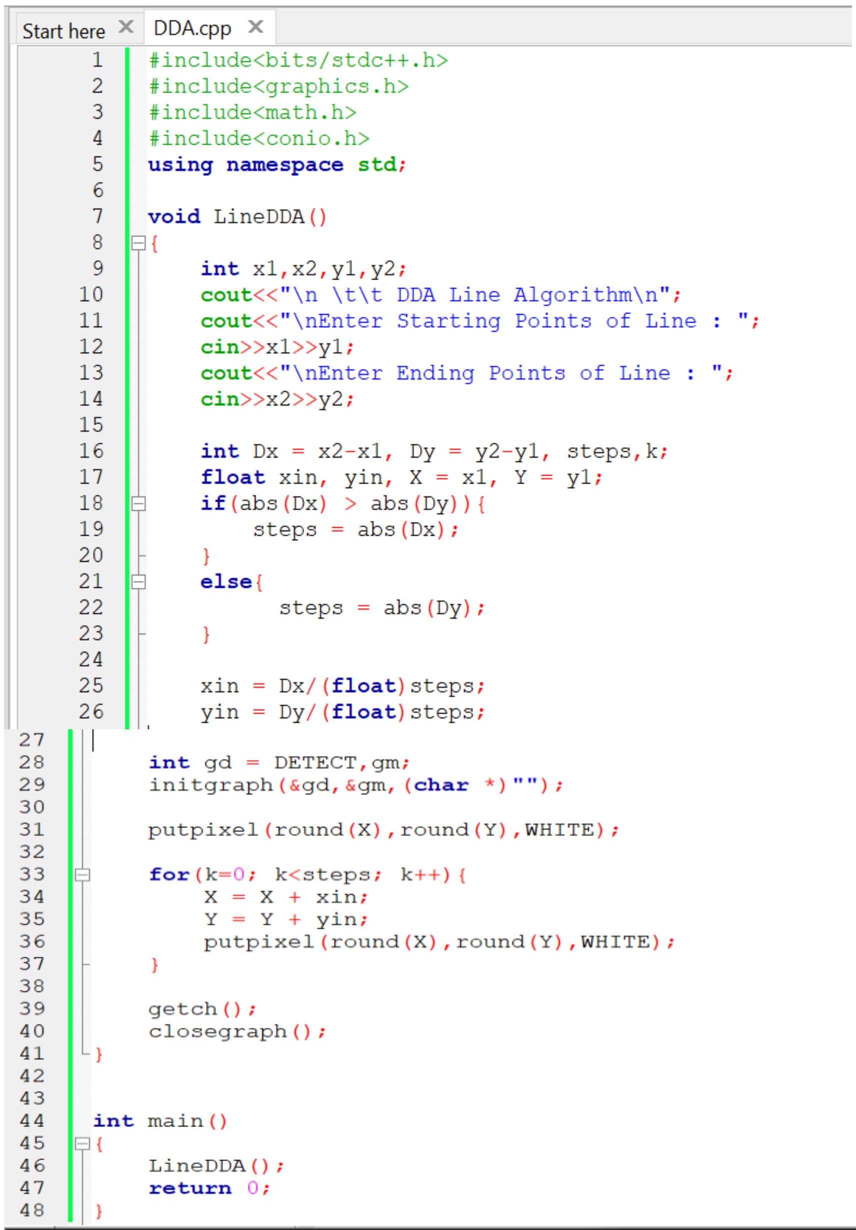
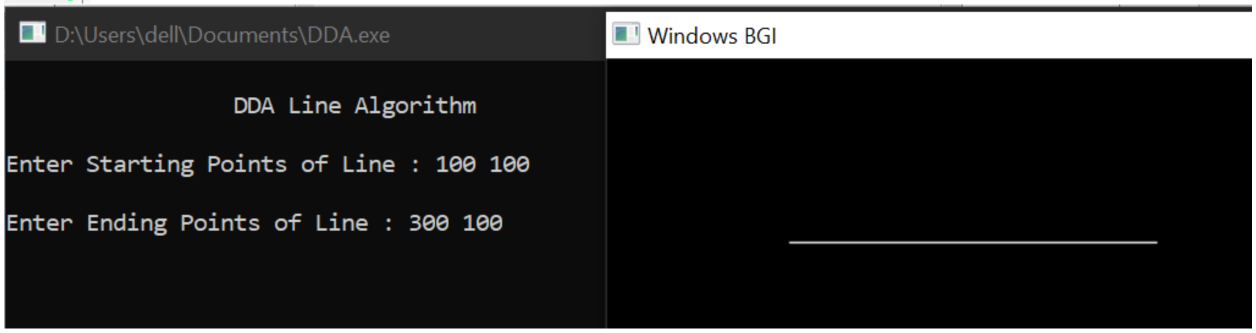


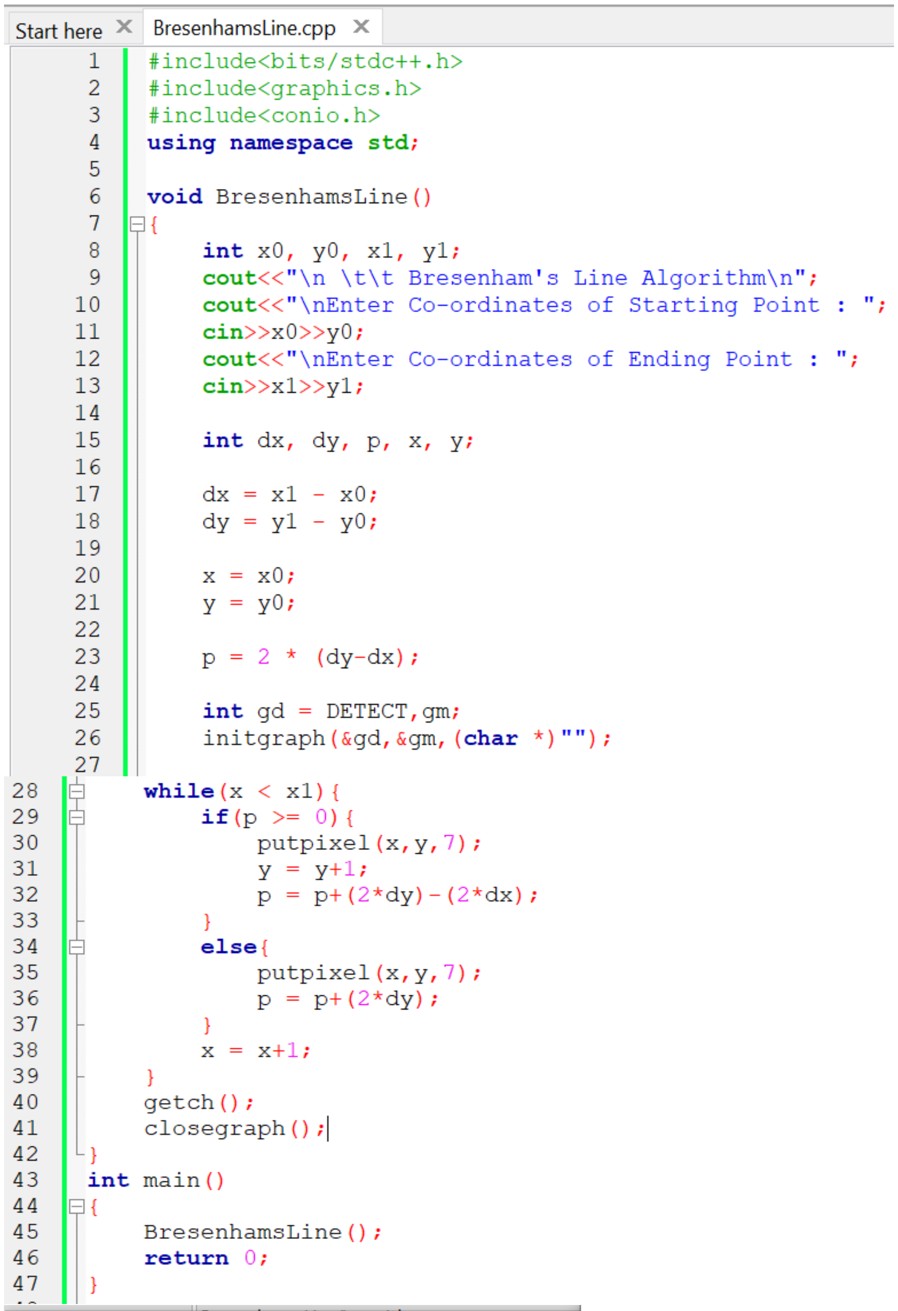
Start here ✕ DDA.cpp ✕

```cpp
#include<bits/stdc++.h>
#include<graphics.h>
#include<math.h>
#include<conio.h>
using namespace std;

void LineDDA()
{
    int x1,x2,y1,y2;
    cout<<"\n \t\t DDA Line Algorithm\n";
    cout<<"\nEnter Starting Points of Line : ";
    cin>>x1>>y1;
    cout<<"\nEnter Ending Points of Line : ";
    cin>>x2>>y2;

    int Dx = x2-x1, Dy = y2-y1, steps,k;
    float xin, yin, X = x1, Y = y1;
    if(abs(Dx) > abs(Dy)){
        steps = abs(Dx);
    }
    else{
          steps = abs(Dy);
    }

    xin = Dx/(float)steps;
    yin = Dy/(float)steps;

    int gd = DETECT,gm;
    initgraph(&gd,&gm,(char *)"");

    putpixel(round(X),round(Y),WHITE);

    for(k=0; k<steps; k++){
        X = X + xin;
        Y = Y + yin;
        putpixel(round(X),round(Y),WHITE);
    }

    getch();
    closegraph();
}


int main()
{
    LineDDA();
    return 0;
}
```

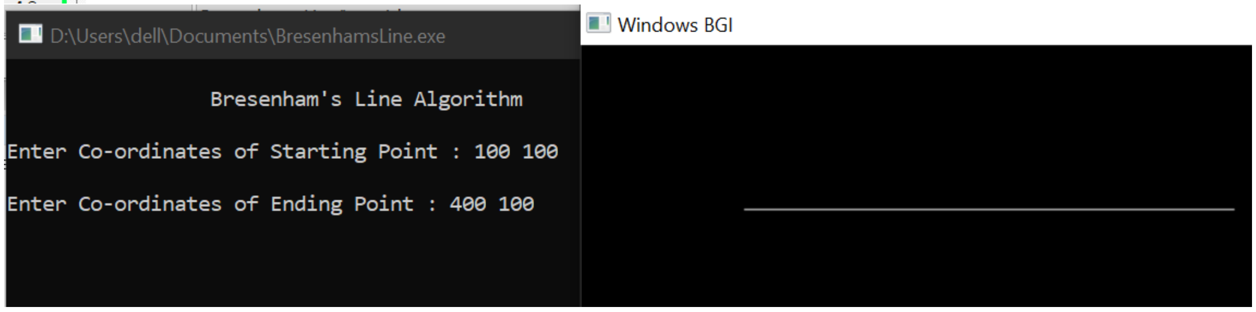


D:\Users\dell\Documents\DDA.exe

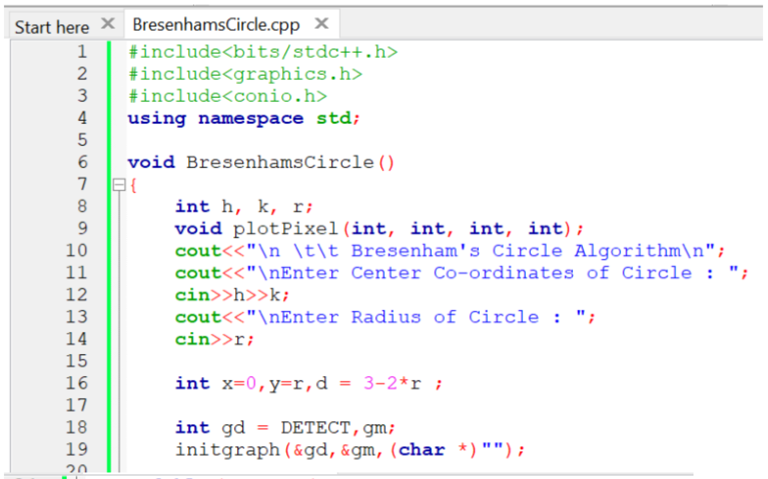
```
                DDA Line Algorithm

Enter Starting Points of Line : 100 100

Enter Ending Points of Line : 300 100
```



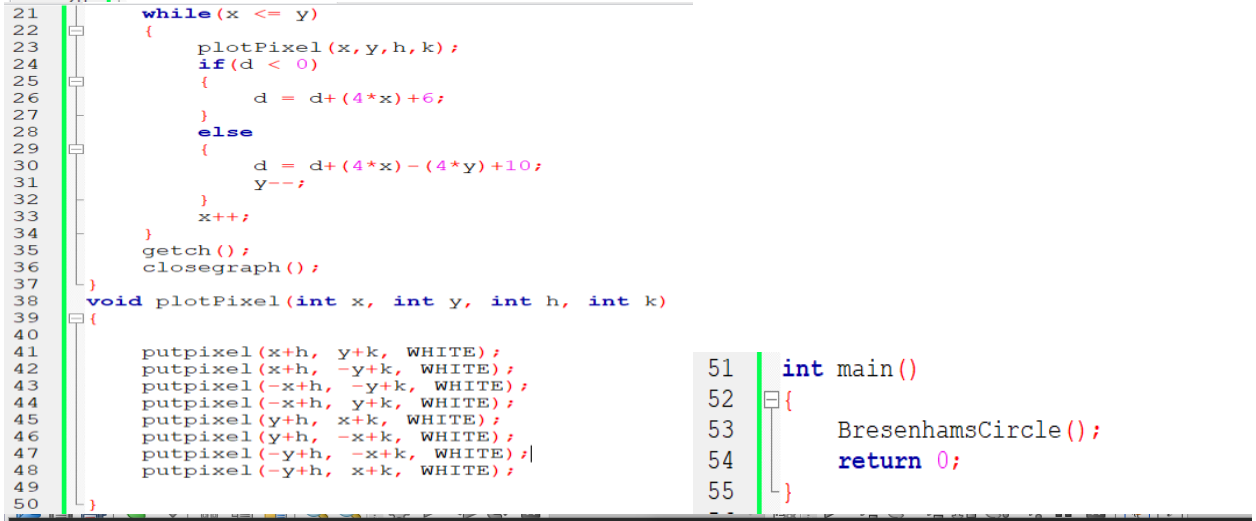
```cpp
#include<bits/stdc++.h>
#include<graphics.h>
#include<conio.h>
using namespace std;

void BresenhamsLine()
{
    int x0, y0, x1, y1;
    cout<<"\n \t\t Bresenham's Line Algorithm\n";
    cout<<"\nEnter Co-ordinates of Starting Point : ";
    cin>>x0>>y0;
    cout<<"\nEnter Co-ordinates of Ending Point : ";
    cin>>x1>>y1;

    int dx, dy, p, x, y;

    dx = x1 - x0;
    dy = y1 - y0;

    x = x0;
    y = y0;

    p = 2 * (dy-dx);

    int gd = DETECT,gm;
    initgraph(&gd,&gm,(char *)"");

    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;
    }
    getch();
    closegraph();
}
int main()
{
    BresenhamsLine();
    return 0;
}
```



```
                 Bresenham's Line Algorithm

Enter Co-ordinates of Starting Point : 100 100

Enter Co-ordinates of Ending Point : 400 100
```

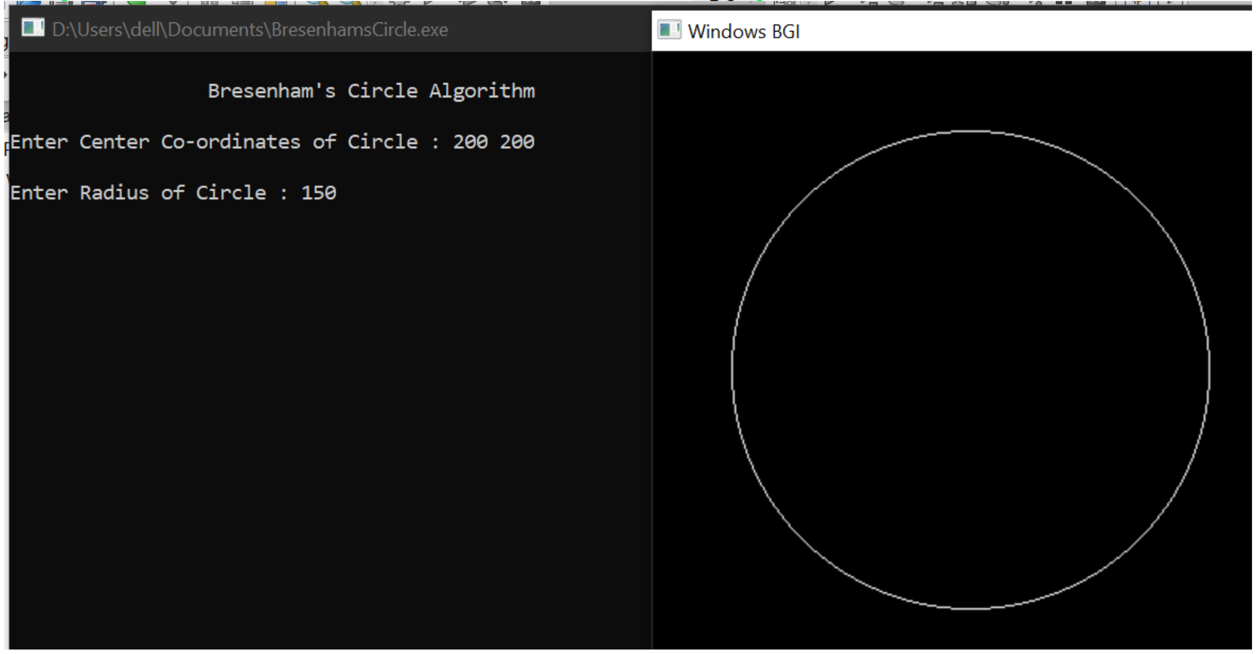


```cpp
#include<bits/stdc++.h>
#include<graphics.h>
#include<conio.h>
using namespace std;

void BresenhamsCircle()
{
    int h, k, r;
    void plotPixel(int, int, int, int);
    cout<<"\n \t\t Bresenham's Circle Algorithm\n";
    cout<<"\nEnter Center Co-ordinates of Circle : ";
    cin>>h>>k;
    cout<<"\nEnter Radius of Circle : ";
    cin>>r;

    int x=0,y=r,d = 3-2*r ;

    int gd = DETECT,gm;
    initgraph(&gd,&gm,(char *)"");
```

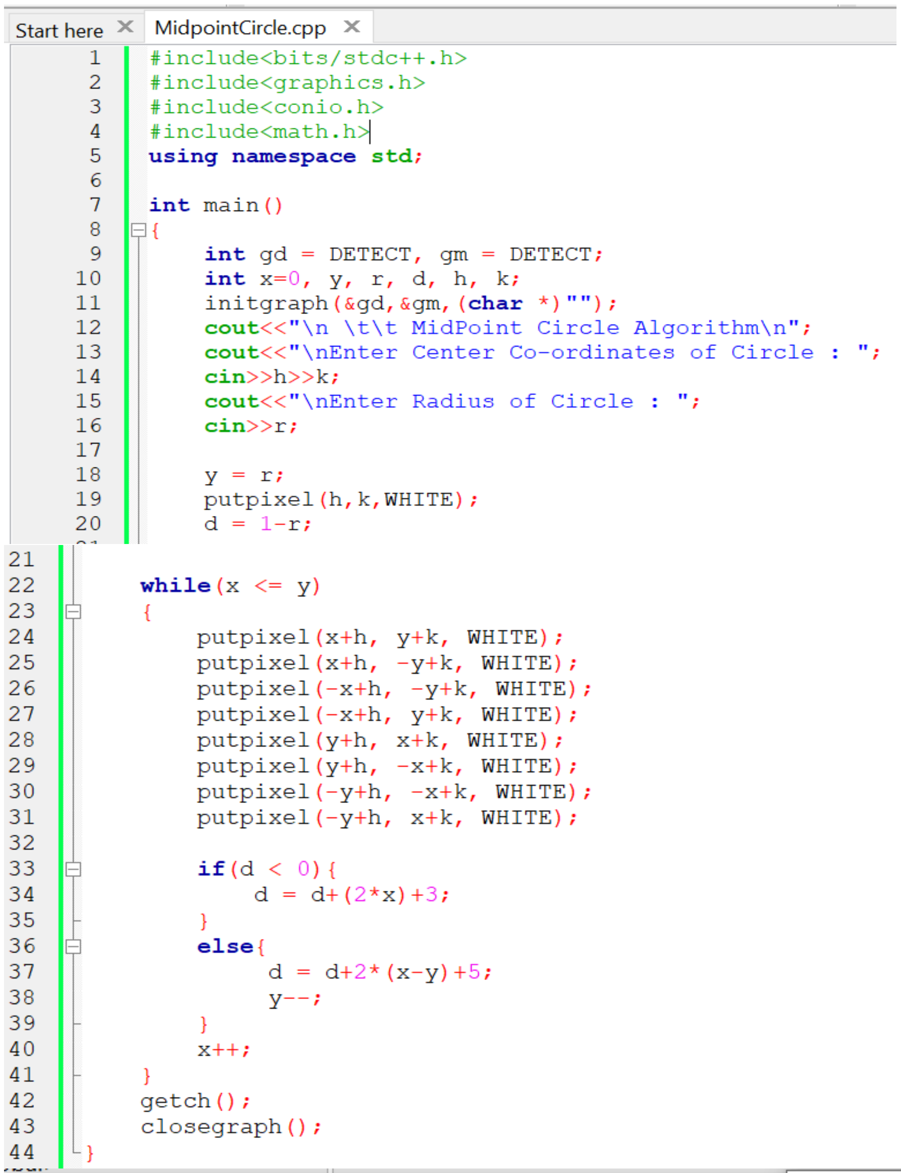


```cpp
    while(x <= y)
    {
        plotPixel(x,y,h,k);
        if(d < 0)
        {
            d = d+(4*x)+6;
        }
        else
        {
            d = d+(4*x)-(4*y)+10;
            y--;
        }
        x++;
    }
    getch();
    closegraph();
}
void plotPixel(int x, int y, int h, int k)
{

    putpixel(x+h, y+k, WHITE);
    putpixel(x+h, -y+k, WHITE);
    putpixel(-x+h, -y+k, WHITE);
    putpixel(-x+h, y+k, WHITE);
    putpixel(y+h, x+k, WHITE);
    putpixel(y+h, -x+k, WHITE);
    putpixel(-y+h, -x+k, WHITE);
    putpixel(-y+h, x+k, WHITE);
}
```

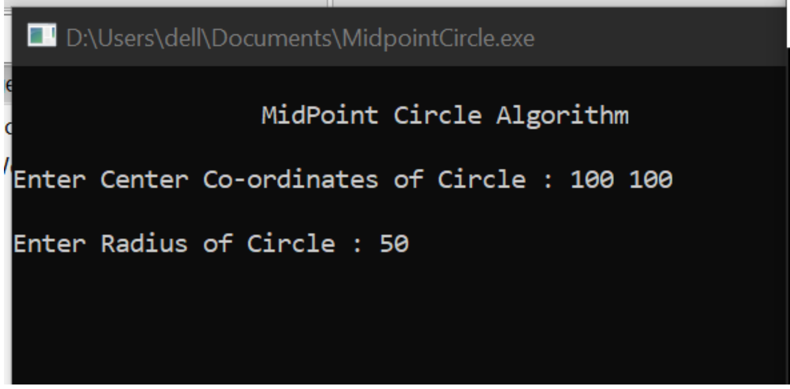
```cpp
int main()
{
    BresenhamsCircle();
    return 0;
}
```



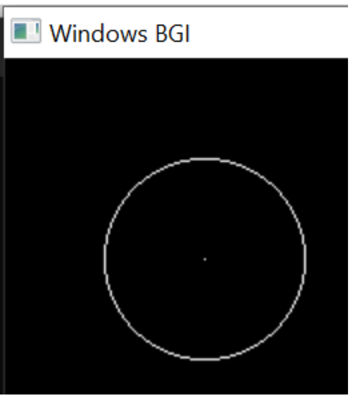
```
                Bresenham's Circle Algorithm

Enter Center Co-ordinates of Circle : 200 200

Enter Radius of Circle : 150
```


Start here ✕ MidpointCircle.cpp ✕

```cpp
#include<bits/stdc++.h>
#include<graphics.h>
#include<conio.h>
#include<math.h>
using namespace std;

int main()
{
    int gd = DETECT, gm = DETECT;
    int x=0, y, r, d, h, k;
    initgraph(&gd,&gm,(char *)"");
    cout<<"\n \t\t MidPoint Circle Algorithm\n";
    cout<<"\nEnter Center Co-ordinates of Circle : ";
    cin>>h>>k;
    cout<<"\nEnter Radius of Circle : ";
    cin>>r;

    y = r;
    putpixel(h,k,WHITE);
    d = 1-r;

    while(x <= y)
    {
        putpixel(x+h, y+k, WHITE);
        putpixel(x+h, -y+k, WHITE);
        putpixel(-x+h, -y+k, WHITE);
        putpixel(-x+h, y+k, WHITE);
        putpixel(y+h, x+k, WHITE);
        putpixel(y+h, -x+k, WHITE);
        putpixel(-y+h, -x+k, WHITE);
        putpixel(-y+h, x+k, WHITE);

        if(d < 0){
            d = d+(2*x)+3;
        }
        else{
             d = d+2*(x-y)+5;
             y--;
        }
        x++;
    }
    getch();
    closegraph();
}
```


D:\Users\dell\Documents\MidpointCircle.exe

```
                  MidPoint Circle Algorithm

Enter Center Co-ordinates of Circle : 100 100

Enter Radius of Circle : 50
```


Windows BGI