

```

#include<bits/stdc++.h>
#include<GL/glut.h>
using namespace std;

int initial=0;
class circle
{
public:
    int cx,cy,r,flag;
    void makecircle(int,int,int,int);
    void Bresenham(int,int,int);
    void DDA(int,int,int);
    void midPoint(int,int,int);
}1;
void circle::makecircle(int x,int y,int cx,int cy){
    glColor3f(0,1,0);
    glVertex2i(cx+x,cy+y);
    glVertex2i(cx+x,cy-y);
    glVertex2i(cx-x,cy+y);
    glVertex2i(cx-x,cy-y);
    glVertex2i(cx+y,cy+x);
    glVertex2i(cx+y,cy-x);
    glVertex2i(cx-y,cy+x);
    glVertex2i(cx-y,cy-x);
}

void circle::Bresenham(int cx,int cy,int r){
    int x=0,y=r,d;
    d=3-2*r;
    while(x<=y){
        if(d>0){
            x++;
            y--;
            d+=4*x-4*y+10;
        }
        else{
            x++;
            d+=4*x+6;
        }
        makecircle(x,y,cx,cy);
    }
    cx=cy=r=initial=0;
}

```

```

void circle::DDA(int xini,int yini ,int rad)
{

```

```

float x1,y1,startx,starty,x2,y2;

x1=rad;
y1=0;
startx=x1;
starty=y1;
int val;
int i=0;
do
{
    val=pow(2,i);
    i++;
}while(val<rad);

float e=1/pow(2,i);

glClear(GL_COLOR_BUFFER_BIT);
glColor3f(0.0,0.0,1.0);
glBegin(GL_POINTS);
do
{
    x2=x1+y1*e;
    y2=y1-e*x2;
    glVertex2f(xini+x2,yini+y2);
    x1=x2;
    y1=y2;
}while((y1-starty)<e|| (startx-x1)>e);
glEnd();
glFlush();
}

```

```

void plot(int x, int y)
{
    glBegin(GL_POINTS);
    glVertex2i(x, y);
    glEnd();
}

```

```

void circle::midPoint(int x,int y,int r)
{
    x = 0;
    y = r;
    float decision = 5/4 - r;
    plot(x, y);

    while (y > x)
    {
        if (decision < 0)
        {
            x++;

```

```

        decision += 2*x+1;
    }
    else
    {
        y--;
        x++;
        decision += 2*(x-y)+1;
    }
    plot(x, y);
    plot(x, -y);
    plot(-x, y);
    plot(-x, -y);
    plot(y, x);
    plot(-y, x);
    plot(y, -x);
    plot(-y, -x);
}

```

```

}

```

```

void display()
{
    if(l.flag==-1){
        glBegin(GL_POINTS);
            l.Bresenham(l.cx,l.cy,l.r);
        glEnd();
    }
    if(l.flag==1){
        glBegin(GL_POINTS);
            l.DDA(l.cx,l.cy,l.r);
        glEnd();
    }
    if(l.flag==0){
        glBegin(GL_POINTS);
            l.midPoint(l.cx,l.cy,l.r);
        glEnd();
    }

    glFlush();
}

```

```

void Init()
{
    glClearColor(0,0,0,0);
    glClear(GL_COLOR_BUFFER_BIT);
    glMatrixMode(GL_PROJECTION);
    gluOrtho2D(0,640,480,0);
}

```

```

void mymouse(int btn, int state,int x, int y)
{
    glPointSize(1.0);
    int a=x,b=y;
    if(btn==GLUT_LEFT_BUTTON&&state==GLUT_DOWN)
    {
        if(initial==0)
        {
            l.cx=a;
            l.cy=b;
            glColor3f(1,0,0);
            glBegin(GL_POINTS);
                glVertex2i(a,b);
            glEnd();
            glFlush();
            initial=1;
        }else
        {
            glColor3f(1,0,0);
            glBegin(GL_POINTS);
                glVertex2i(a,b);
            glEnd();
            glFlush();
            l.r=sqrt(pow(a-l.cx,2)+pow(b-l.cy,2));
            initial=0;
            display();
        }
    }
}

```

```

void options(int id)
{
    switch(id)
    {
        case 1:
            l.flag=1;
            break;
        case 2:
            l.flag=-1;
            break;
        case 3:
            l.flag=0;
            break;
        case 4:
            glClear(GL_COLOR_BUFFER_BIT);
            glFlush();
            break;
    }
}

```

```
int main(int argc,char* argv[])
{
    glutInit(&argc,argv);
    glutInitWindowPosition(100,100);
    glutInitWindowSize(640,480);
    glutCreateWindow("MIDPOINT,BRESENHAM,DDA");

    Init();

    glutDisplayFunc(display);
    glutMouseFunc(mymouse);

    glutCreateMenu(options);
        glutAddMenuEntry("DDA",1);
        glutAddMenuEntry("Bresenham",2);
        glutAddMenuEntry("MidPoint",3);
        glutAddMenuEntry("Clear",4);
    glutAttachMenu(GLUT_RIGHT_BUTTON);
    glutMainLoop();
    return 0;
}
```