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

void myArc2(int xc,int yc,int startangle,int stopangle,int radius,int theta,int omega,vector<pair<int,int> > &pts)
{

    double x=radius*cos((3.14159*startangle)/180),y=radius*sin((3.14159*startangle)/180);

    for(int X=startangle;X%360!=stopangle%360;X+=1)
    {
        x=xc+radius*cos((3.14159*X)/180);
        y=yc-radius*sin((3.14159*X)/180);
        int Tx=xc+(x-xc)*cos(omega*3.14/180)-(y-yc)*sin(omega*3.14/180);
        int Ty=yc+(x-xc)*sin(omega*3.14/180)+(y-yc)*cos(omega*3.14/180);
        pts.push_back({Tx,Ty});
    }
}
void rangoli(int Cx,int Cy,int radius,int theta=0,int omega=0,int color=WHITE)
{
    int F=radius*0.50,S=radius*0.87;
    pair <int,int> Cen[10]={
        {Cx+F,Cy+S},{Cx-F,Cy+S},
        {Cx-radius,Cy},{Cx-F,Cy-S},{Cx+F,Cy-S},{Cx+radius,Cy}
    };


    for (int i=0;i<6;i++)
    {
        int temp=Cx+(Cen[i].first-Cx)*cos(omega*3.14/180)-(Cen[i].second-Cy)*sin(omega*3.14/180);
Cen[i].second=Cy+(Cen[i].first-Cx)*sin(omega*3.14/180)+(Cen[i].second-Cy)*cos(omega*3.14/180);
        Cen[i].first=temp;
    }

    vector <pair <int,int> > pts;
    myArc2(Cen[0].first,Cen[0].second,(+60),(+180),radius,theta,omega,pts);
    myArc2(Cen[1].first,Cen[1].second,0,+120,radius,theta,omega,pts);
    myArc2(Cen[2].first,Cen[2].second,+300,+60,radius,theta,omega,pts);
    myArc2(Cen[3].first,Cen[3].second,240,+360,radius,theta,omega,pts);
    myArc2(Cen[4].first,Cen[4].second,+180,+300,radius,theta,omega,pts);
    myArc2(Cen[5].first,Cen[5].second,+120,+240,radius,theta,omega,pts);
    srand(17);
    random_shuffle(pts.begin(),pts.end());///less flickering
    circle(Cx,Cy,radius);
    int COL=YELLOW;
    for (auto x:pts)
    {
        putpixel(x.first,x.second,COL);
    }
    delay(1);

}
int main()
{
    int gd = DETECT, gm;
    ///initgraph(&gd, &gm, "");
    initwindow(1700,1700);

    int a=120,b=80,C=350;
    int radiusRangoli=30;
    int theta=0;double omega=0;

    int Cx=C,Cy=C;
    int x0=Cx+a,y0=Cy;
    while(1)
    for (theta=0;theta<360;theta+=1)
    {
        ellipse(C,C,0,360,a,b);

        x0=C+a*cos(theta*3.14/180);y0=C+b*sin(theta*3.14/180);
        double R=sqrt((x0-C)*(x0-C)+(y0-C)*(y0-C));
        double d_omega=R/radiusRangoli;
        omega+=d_omega;
        omega=fmod(omega,360);
        cout<<omega<<" "<<theta<<"\n";

        Cx=x0+radiusRangoli*cos(theta*3.14/180);
        Cy=y0+radiusRangoli*sin(theta*3.14/180);
        rangoli(Cx,Cy,radiusRangoli,theta,omega);
```

```c
        clearviewport();
    }


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