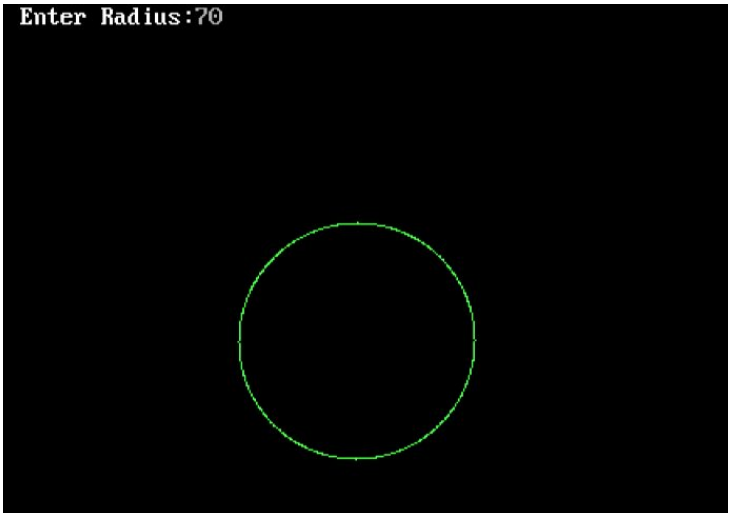
Q5 WAP to make a circle by using Mid-Point circle algorithm

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
#include<math.h>
void main()
{
 int gdriver=DETECT,gmode,errorcode,tmp,i=1,rds;
 float st_x,st_y,x1,x2,y1,y2,ep;
 initgraph(&gdriver,&gmode,"C:\\TC\\BGI");
       printf("Enter Radius:");
 scanf("%d",&rds);
 while(rds>pow(2,i))
  i++;
   ep=1/pow(2,i);
 x1=rds;  y1=0;
 st_x=rds; st_y=0;
 do
 { x2=x1+(y1*ep);
  y2=y1-(x2*ep);
  putpixel(x2+200,y2+200,10);
  x1=x2;
  y1=y2;
 }while((y1-st_y)<ep || (st_x-x1)>ep);
    getch();
}
```

OUTPUT

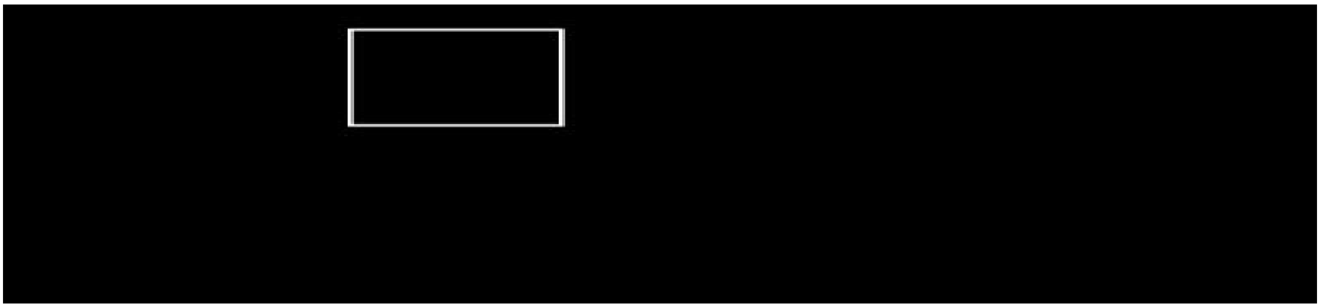


```
Enter Radius:70
```

Q6 WAP to make a rectangle by using DDA line algorithm

```
#include<stdio.h>
#include<graphics.h>
#include<conio.h>
#include<math.h>
#define ROUND(x) ((int) (x+0.5))

//DDA Function
void ddaline(int x1, int y1, int x2, int y2, int c)
{
   // calculate dx & dy
   int dx = x2 - x1;
   int dy = y2 - y1;

   // calculate steps required for generating pixels
   int steps;
   if(abs(dx) > abs(dy))
         steps = abs(dx);
   else
         steps = abs(dy);

   // calculate increment in x & y for each steps
   float x_inc = dx / (float)steps;
   float y_inc = dy / (float)steps;

   // Put pixel for each step
   float x = x1;
   float y = y1;
   putpixel(x,y,c);
   for (int i = 0; i < steps; ++i)
   {
         x += x_inc;
         y += y_inc;
         putpixel(ROUND(x), ROUND(y),c);
   }
}

// Driver program
void main()
{
   int gd = DETECT, gm;

   // Initialize graphics function
   initgraph (&gd, &gm, "C:\\TC\\BGI");
   ddaline(10,10,90,10,111);
   ddaline(10,10,10,50,111);
   ddaline(90,10,90,50,111);
   ddaline(10,50,90,50,111);
   getch();
}
```

OUTPUT



Q7 WAP to make flying colored balloons.

```
#include<iostream.h>
#include<graphics.h>
#include<conio.h>
#include<dos.h>
void main(){
        int gd = DETECT, gm;
        initgraph(&gd, &gm, "C:\\TC\\BGI");
        for(int j=0;j<5;j++)
        {
        for(int i=0;i<600;i++)
        {
        setfillstyle(SOLID_FILL,MAGENTA);
        circle(50,390-i,50);
        floodfill(50,390-i,WHITE);
        setfillstyle(SOLID_FILL,GREEN);
        circle(90+i,390-2*i,50);
        floodfill(90+i,390-2*i,WHITE);
        setfillstyle(SOLID_FILL,BLUE);
        circle(135+2*i,393-i,50);
        floodfill(130+2*i,390-i,WHITE);
        setfillstyle(SOLID_FILL,WHITE);
        circle(195+2*i,393-3*i,50);
        floodfill(195+2*i,393-3*i,WHITE);

        delay(5);
        cleardevice();
        }
        }
        getch();
}
```

OUTPUT