ITM UNIVERSITY
GWALIOR • MP • INDIA
"CELEBRATING DREAM

Lab File

Computer graphics

MCA (202/402)

SUBMITTED BY
Prabhat Singh Bhadoriya
MCAN1CA20015

SUBMITTED TO
Mr. Pankaj Gugnani
Assistant prof. (CSA)

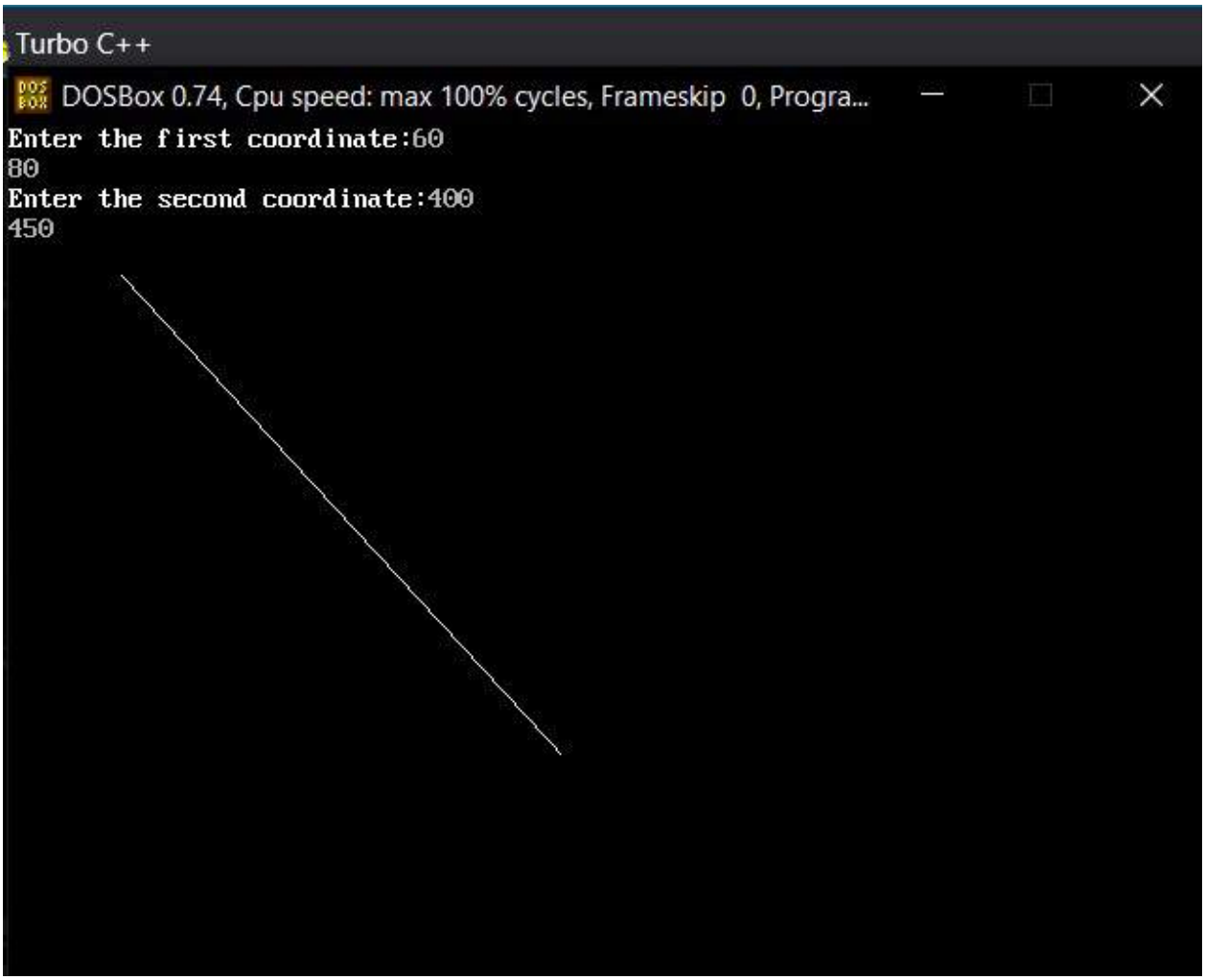
Index

S.no	Program name	Date
1	Write a program to implement DDA line drawing algorithm	
2	Write a program to implement Bresenham's line drawing algorithm	
3	Write a program to implement Bresenham's circle drawing algorithm	
4	Write a program to implement mid-point circle drawing algorithm.	
5	Write a program to perform two-dimensional translation of an object (preferably rectangle). The translation vectors should be entered at console during runtime	
6	Write a program to draw different geometric shapes (line, square, rectangle, triangle, eclipse, etc.) using built-in library functions	
7	Write a program to implement flood-fill algorithm for region filling	
8	Write a program to implement moving car	
9	Write a program to draw a national flag	
10	A man walking with a umbrella on hand in rain	

1. **Write a program to implement DDA line drawing algorithm.**

```
#include <iostream.h>
#include <conio.h>
#include <math.h>
#include <dos.h>
#include <graphics.h>
void main()
{
    float x1, x2, y1, y2, x, y, xi, yi;
    int l;
    clrscr();
    int gd = DETECT, gm;
    initgraph(&gd, &gm, "C:\\TURBOC3\\BGI");
    cout << "Enter the first coordinate:";
    cin >> x1 >> y1;
    cout << "Enter the second coordinate:";
    cin >> x2 >> y2;
    l = abs(x2 - x1);
    if (abs(y2 - y1) > l)
    {
        l = abs(y2 - y1);
    }
    xi = (x2 - x1) / l;
    yi = (y2 - y1) / l;
    x = x1 + 0.5;
    y = y1 + 0.5;
    putpixel(x, y, WHITE);
    for (int i = 0; i < l; i++)
    {
        x = x + xi;
        y = y + yi;
        putpixel(x, y, WHITE);
        delay(10);
    }
    getch();
    closegraph();
}
```

OUTPUT:

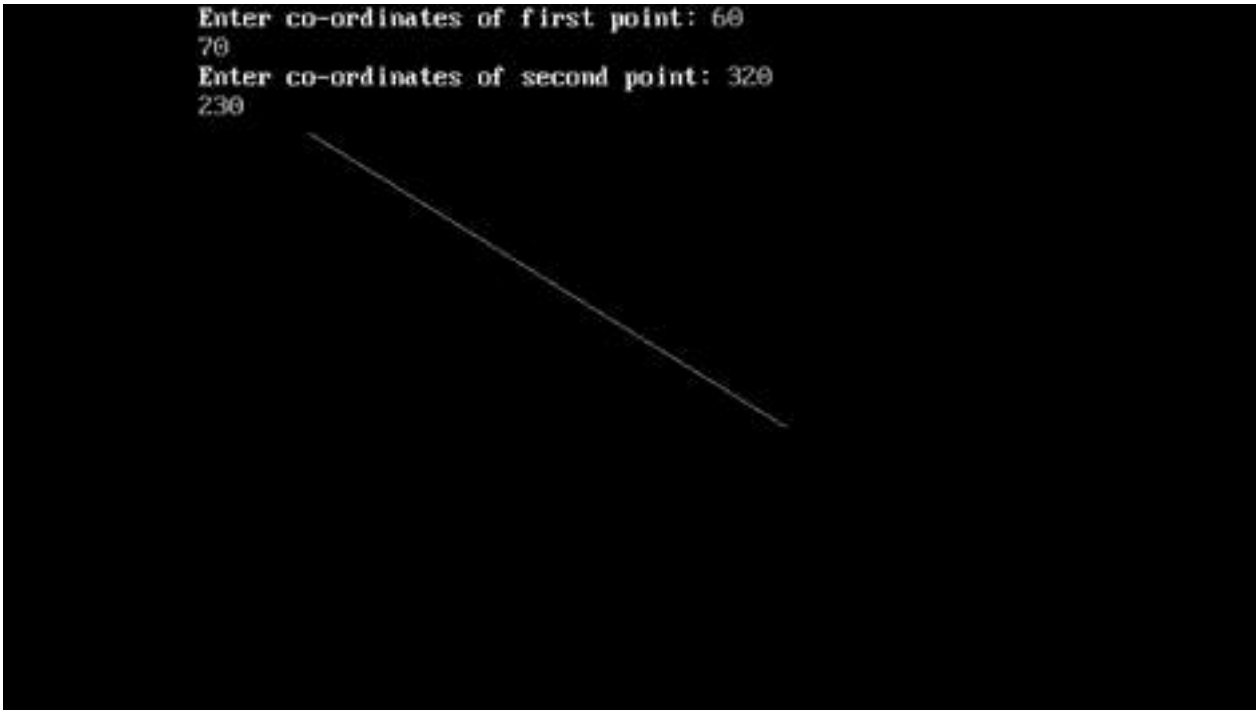

Turbo C++
DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Progra...
Enter the first coordinate:60
80
Enter the second coordinate:400
450

2. **Write a program to implement Bresenham's line drawing algorithm**

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

void drawline(int x0, int y0, int x1, int y1)
{
   int dx, dy, p, x, y;

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

      x=x0;
      y=y0;

      p=2*dy-dx;

      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;
      }
}
```

```
int main()
{
	int gdriver=DETECT, gmode, error, x0, y0, x1, y1;
	initgraph(&gdriver, &gmode, "C:\\TURBOC3\\BGI");

	cout<<"Enter co-ordinates of first point: ";
	cin>>x0>>y0;

	cout<<"Enter co-ordinates of second point: ";
	cin>>x1>>y1;
	drawline(x0, y0, x1, y1);
	getch();
	return 0;
}
```

OUTPUT:

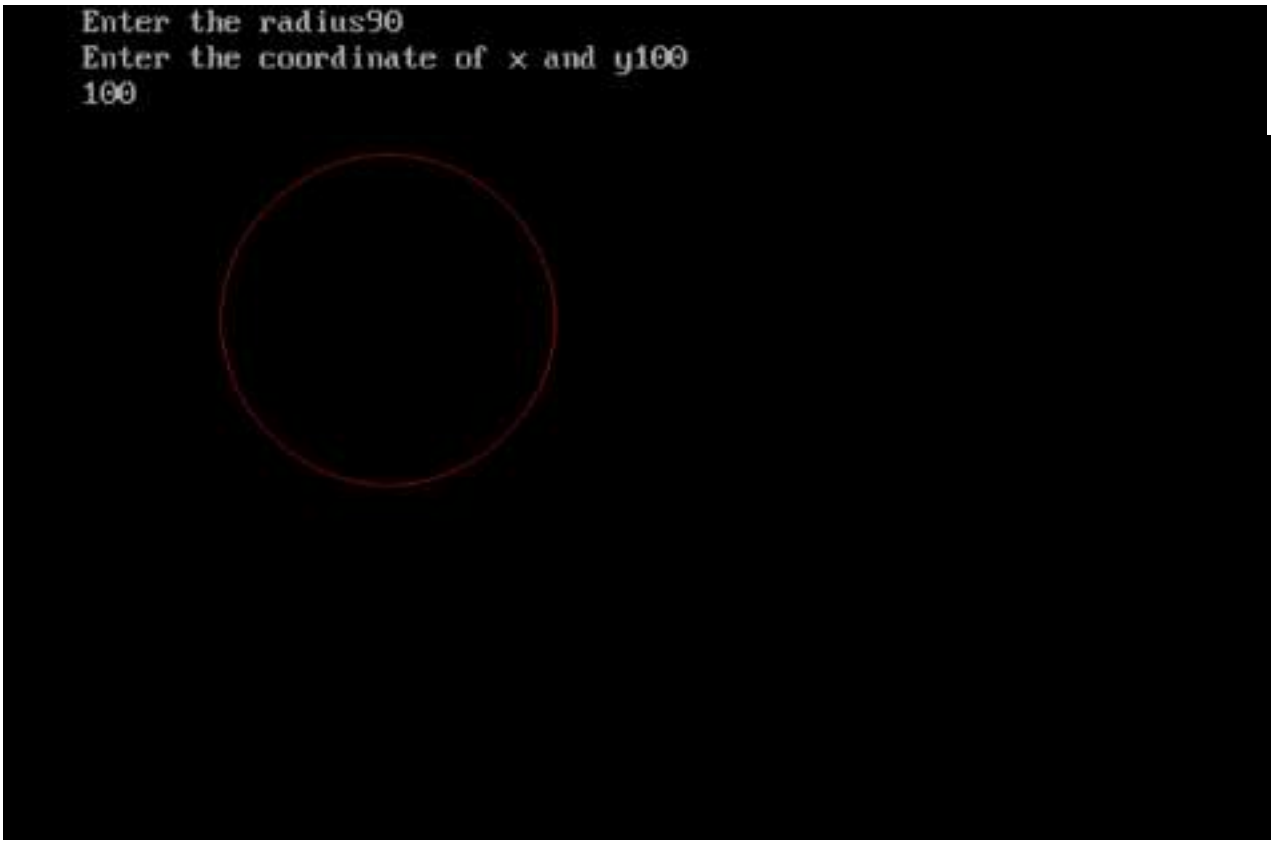


3. **Write a program to implement Bresenham's circle drawing algorithm**

```
#include <stdio.h>
#include <conio.h>
#include <dos.h>
#include <graphics.h>

void drawCircle(int xc, int yc, int x, int y)
{
	putpixel(xc+x, yc+y, RED);
	putpixel(xc-x, yc+y, RED);
	putpixel(xc+x, yc-y, RED);
	putpixel(xc-x, yc-y, RED);
	putpixel(xc+y, yc+x, RED);
	putpixel(xc-y, yc+x, RED);
	putpixel(xc+y, yc-x, RED);
	putpixel(xc-y, yc-x, RED);
}

void circleBres(int xc, int yc, int r)
{
	int x = 0, y = r;
	int d = 3 - 2 * r;
	drawCircle(xc, yc, x, y);
	while (y >= x)
	{
		x++;
		if (d > 0)
		{
			y--;
			d = d + 4 * (x - y) + 10;
		}
		else
			d = d + 4 * x + 6;
```

```
            drawCircle(xc, yc, x, y);
            delay(50);
        }
}
int main()
{
        int xc = 50, yc = 50, r = 30;
        int gd = DETECT, gm;
        initgraph(&gd, &gm, "C:\\TURBOC3\\BGI");
        circleBres(xc, yc, r);
        getch();
        return 0;
}
```

OUTPUT:

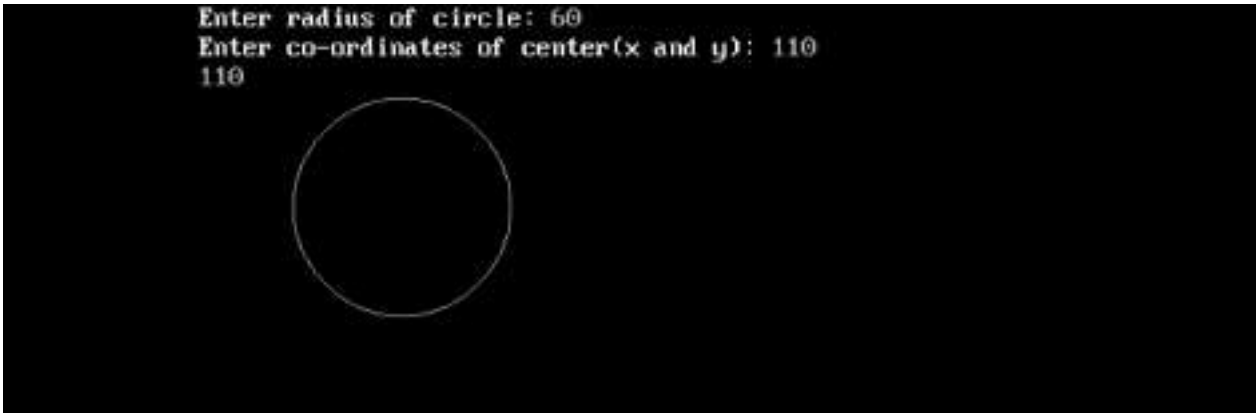


4. **Write a program to implement mid-point circle drawing algorithm.**

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

void drawcircle(int x0, int y0, int radius)
{
    int x = radius;
    int y = 0;
    int err = 0;

    while (x >= y)
    {
        putpixel(x0 + x, y0 + y, 7);
        putpixel(x0 + y, y0 + x, 7);
        putpixel(x0 - y, y0 + x, 7);
        putpixel(x0 - x, y0 + y, 7);
        putpixel(x0 - x, y0 - y, 7);
        putpixel(x0 - y, y0 - x, 7);
        putpixel(x0 + y, y0 - x, 7);
        putpixel(x0 + x, y0 - y, 7);

        if (err <= 0)
        {
            y += 1;
            err += 2*y + 1;
        }

        if (err > 0)
        {
            x -= 1;
            err -= 2*x + 1;
```

```
        }
    }
}

int main()
{
        int gdriver=DETECT, gmode, error, x, y, r;
        initgraph(&gdriver, &gmode, "c:\\turboc3\\bgi");

        cout<<"Enter radius of circle: ";
        cin>>r;

        cout<<"Enter co-ordinates of center(x and y): ";
        cin>>x>>y;
        drawcircle(x, y, r);
        getch();
        return 0;
}
```

OUTPUT:

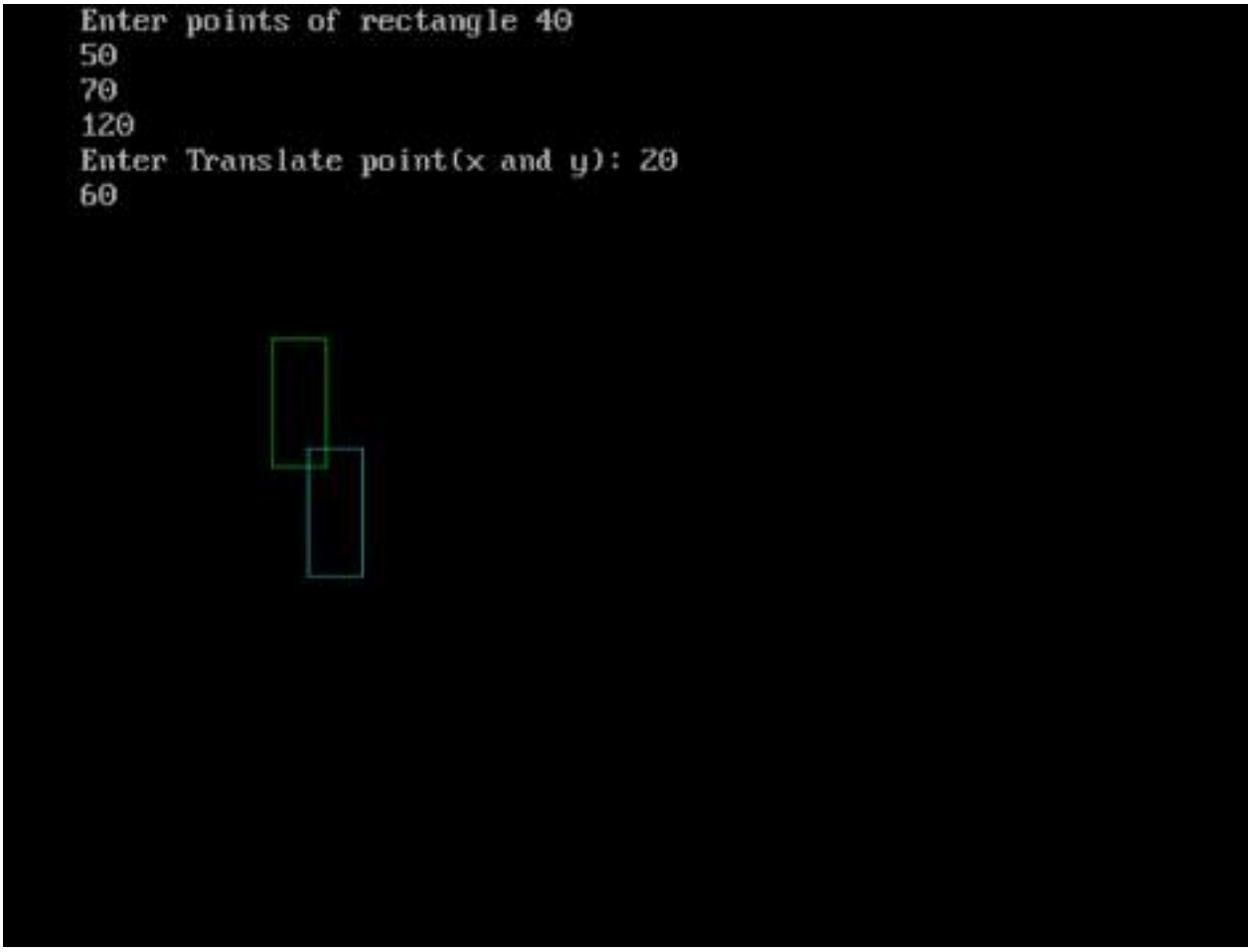


5. **Write a program to perform two-dimensional translation of an object (preferably rectangle). The translation vectors should be entered at console during runtime.**

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

void translateRectangle ( int P[][2], int T[])
{

	int gd = DETECT, gm, errorcode;
	initgraph (&gd, &gm, "c:\\turboc3\\bgi");
	setcolor (2);

	rectangle (P[0][0], P[0][1], P[1][0], P[1][1]);

	P[0][0] = P[0][0] + T[0];
	P[0][1] = P[0][1] + T[1];
	P[1][0] = P[1][0] + T[0];
	P[1][1] = P[1][1] + T[1];

	setcolor(3);
	rectangle (P[0][0], P[0][1], P[1][0], P[1][1]);
}

int main()
{

	int a,b,c,d,x,y;
	cout<<"Enter points of rectangle ";
	cin>>a>>b>>c>>d;
	cout<<"Enter Translate point(x and y): ";
	cin>>x>>y;
	int P[2][2] = {a,b,c,d};
```

```
        int T[] = {x,y};
        translateRectangle (P, T);
        getch();
        return 0;
}
```

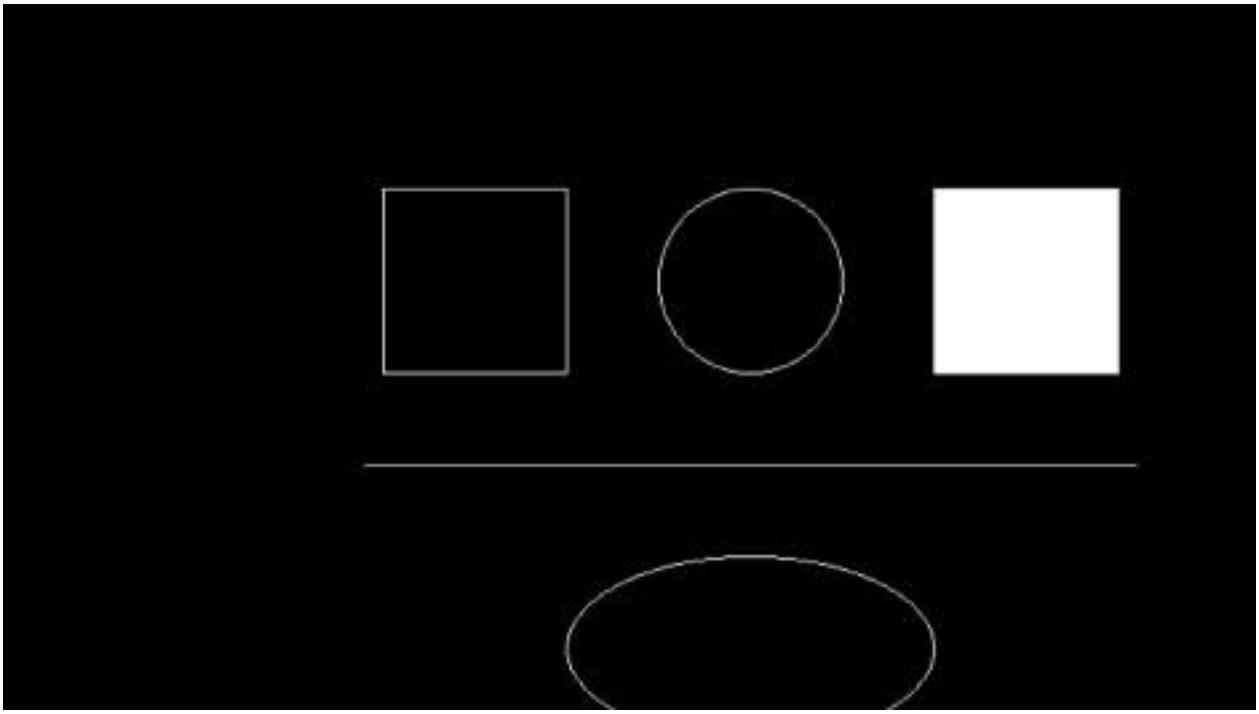
OUTPUT:



6. **Write a program to draw different geometric shapes (line, square, rectangle, triangle, eclipse, etc.) using built-in library functions.**

```
#include<iostream.h>
#include<graphics.h>
#include<conio.h>
main()
{
  int gd = DETECT,gm,left=100,top=100,right=200,bottom=200,x=
300,y=150,radius=50;

  initgraph(&gd, &gm, "C:\\TURBOC3\\BGI");

  rectangle(left, top, right, bottom);
  circle(x, y, radius);
  bar(left + 300, top, right + 300, bottom);
  line(left - 10, top + 150, left + 410, top + 150);
  ellipse(x, y + 200, 0, 360, 100, 50);
  //outtextxy(left + 100, top + 325, "My first C graphics program");

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

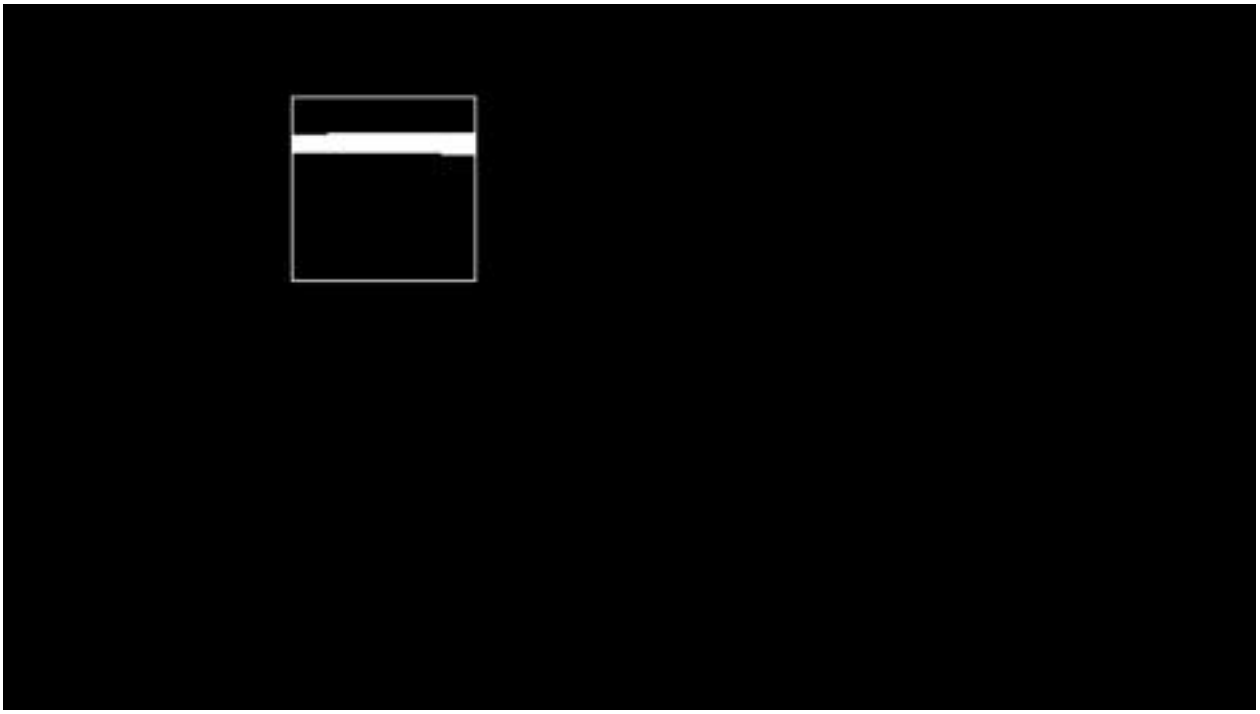
OUTPUT:



7. **Write a program to implement flood-fill algorithm for region filling.**

```
#include<stdio.h>
#include<graphics.h>
#include<dos.h>
#include<conio.h>
void floodfill(intx,inty,intold,intnewcol)
{
        int current;
        current=getpixel(x,y);
        if(current==old)
        {
                delay(5);
                putpixel(x,y,newcol);
                floodfill(x+1,y,old,newcol);
                floodfill(x-1,y,old,newcol);
                floodfill(x,y+1,old,newcol);
                floodfill(x,y-1,old,newcol);
                floodfill(x+1,y+1,old,newcol);
                floodfill(x-1,y+1,old,newcol);
                floodfill(x+1,y-1,old,newcol);
                floodfill(x-1,y-1,old,newcol);
        }
}
void main()
{
        intgd=DETECT,gm;
        initgraph(&gd,&gm,"C:\\TURBOC3\\BGI");
        rectangle(50,50,150,150);
        floodfill(70,70,0,15);
        getch();
        closegraph();
}
```

OUTPUT:

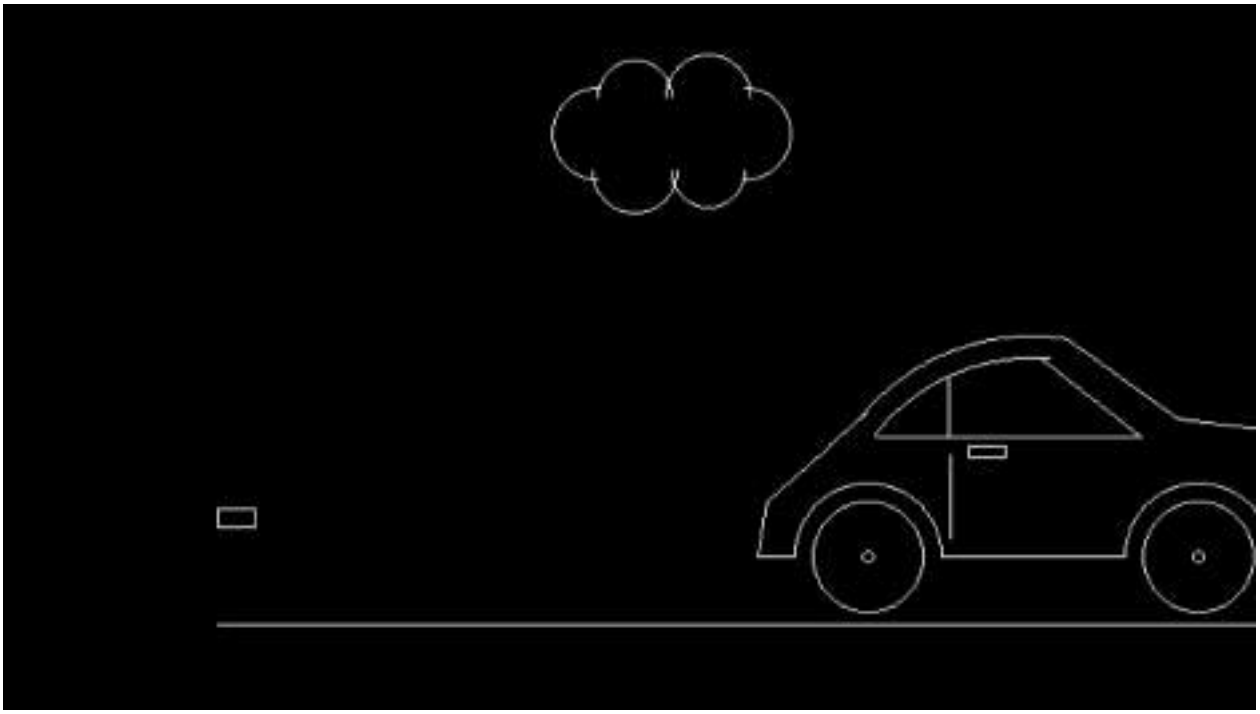


8. **Write a program to implement moving car**

```
#include<iostream.h>
#include<conio.h>
#include<graphics.h>
#include<dos.h>
void main()
{
clrscr();
int gd = DETECT, gm;
initgraph(&gd, &gm, "C:\\TURBOC3\\BGI");
for(int i =0; i<=600; i++){
circle(100+i,300,30);//1st wheel
circle(100+i,300,3);
line(140+i,300,240+i,300);//base of car
circle(100+i,300,3);
circle(280+i,300,30);//2nd Wheel
circle(280+i,300,3);
arc(100+i,300,0,180,40);//1st wheel marguard
arc(280+i,300,0,180,40);//2nd wheel marguard
arc(190+i,300,80,140,120);//roof
arc(190+i,300,85,142,108);
line(210+i,182,270+i,225);//front mirror
line(270+i,225,350+i,235);// front horizontal line
line(350+i,235,350+i,300);// front vertical line
arc(350+i,235,175,270,12);//Headlight
line(342+i,265,342+i,285);
line(342+i,265,357+i,265);
line(342+i,285,357+i,285);
line(357+i,262,357+i,288);
line(350+i,300,320+i,300);// base joining 2nd margaurd
line(60+i,300,40+i,300);
line(40+i,300,45+i,270);
line(45+i,270,100+i,222);
line(250+i,235,106+i,235);
```

```
line(195+i,192,250+i,235);
line(145+i,290,145+i,245);
line(145+i,235,145+i,203);
line(155+i,240,175+i,240);
line(155+i,246,175+i,246);
line(155+i,240,155+i,246);
line(175+i,240,175+i,246);
line(10,337,600,337);//road
line(10,10+i,30,10+i);
line(10,10+i,10,20+i);line(30,10+i,30,20+i);
line(10,20+i,30,20+i);
/* Cloud */
arc(500-i,50,0,180,20);
arc(540-i,50,0,180,23);
arc(500-i,90,180,360,23);
arc(540-i,90,180,360,20);
arc(480-i,70,90,270,25);
arc(560-i,70,270,90,25);
delay(20);
cleardevice();
}
outtextxy(200,200,"The end");
getch();
closegraph();
}
```

OUTPUT:

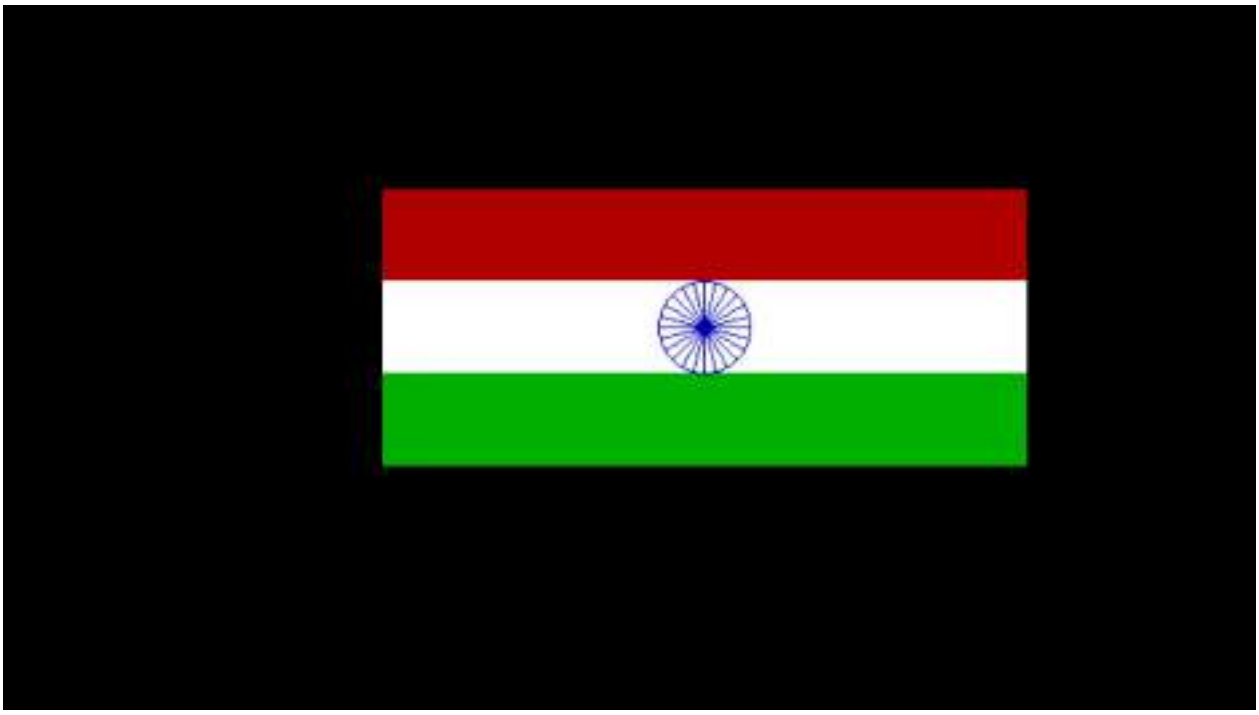


9. **Write a program to draw a national flag.**

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
#include<math.h>
int main()
{
	int gd,gm;
	int r,i,a,b,x,y;
	float PI=3.14;

	detectgraph(&gd,&gm);
	initgraph(&gd,&gm,"C:\\TURBOC3\\BGI");

	setcolor(RED);
	rectangle(100,100,450,150);
	setfillstyle(SOLID_FILL,RED);
	floodfill(101,101,RED);

	setcolor(WHITE);
	rectangle(100,150,450,200);
	setfillstyle(SOLID_FILL,WHITE);
	floodfill(101,151,WHITE);

	setcolor(GREEN);
	rectangle(100,200,450,250);
	setfillstyle(SOLID_FILL,GREEN);
	floodfill(101,201,GREEN);

	a=275;	//center
	b=175;	//center
	r=25;	//radius
	setcolor(BLUE);
	circle(a,b,r);
```

```
        //spokes
        for(i=0;i<=360;i=i+15)
        {
                x=r*cos(i*PI/180);
                y=r*sin(i*PI/180);
                line(a,b,a+x,b-y);
        }

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

OUTPUT:



10. **A man walking with a umbralla on hand in hand**

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
#include<stdlib.h>
#include<dos.h>
void main()
{
int gd=DETECT,gm;
int rhx,rhy,j,i;
clrscr();
initgraph(&gd,&gm,"C:\\TURBOC3\\BGI");
for(i=0;i<500;i+=5)
{
line(20,380,580,380); //platform

if(i%2==0)
{
line(25+i,380,35+i,340); //leftleg
line(45+i,380,35+i,340);//right leg
line(35+i,310,25+i,330);//left hand
delay(20);
}
else
{
line(35+i,380,35+i,340);
line(35+i,310,40+i,330);
delay(20);
}

line(35+i,340,35+i,310); //body
circle(35+i,300,10); //head
line(35+i,310,50+i,330); // hand
line(50+i,330,50+i,280); //umbrella stick
```

```
line(15+i,280,85+i,280); //umbrella right

arc(50+i,280,0,180,35); //umbrella body
arc(55+i,330,180,360,5);//umbrella handle
rhx=getmaxx();
rhy=getmaxy();

for(j=0;j<100;j++)
{
outtextxy(random(rhx),random(rhy-50),".");

setcolor(WHITE);
}
delay(150);
cleardevice();
}

getch();
}
```

OUTPUT: