

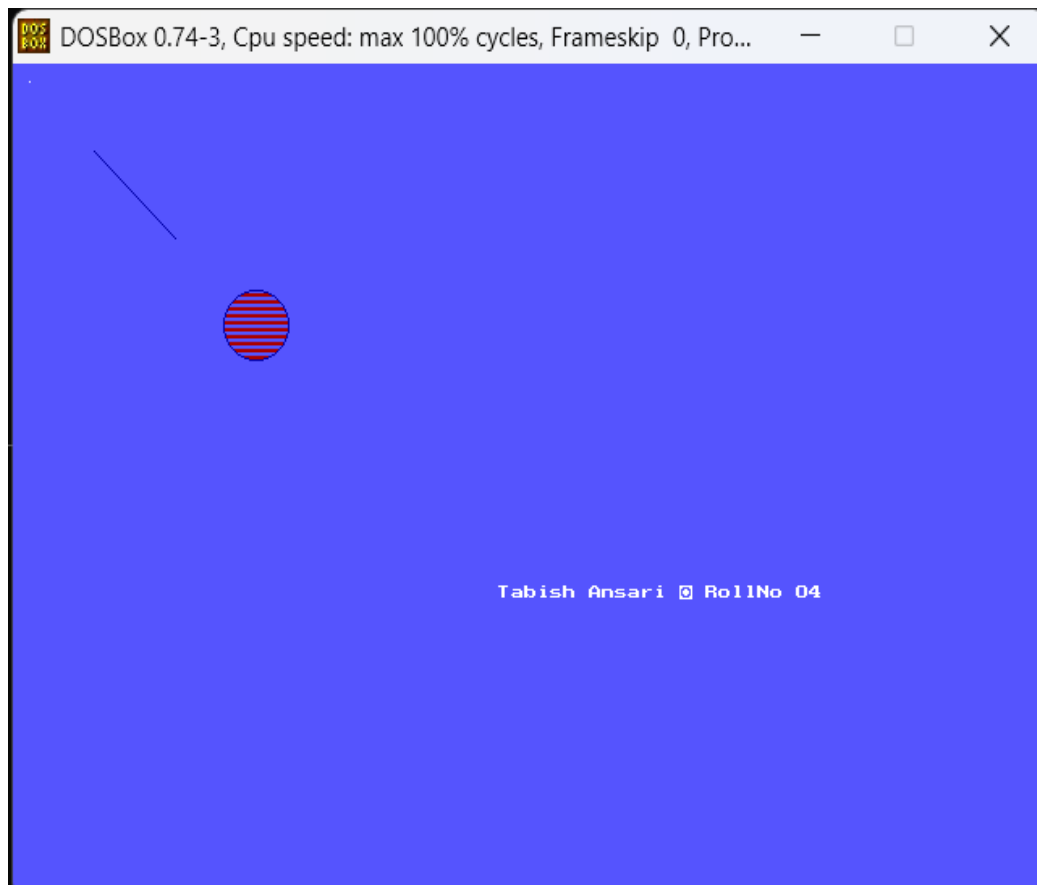
Experiment-1:

AIM: Study and enlist the basic function used for graphics in C language. Give example for each of them.

CODE:

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
void main(){
int gd = DETECT,gm,color;
initgraph(&gd,&gm,"C:\\\\TURBOC3\\\\BGI");
outtextxy(300,300,"Tabish Ansari \n RollNo 04");
setcolor(1); //darkblue
setbkcolor(9); //lightblue
line(50,50,100,100);
circle(150,150,20);
setfillstyle(2,RED);
floodfill(150,150,1);
putpixel(10,10,WHITE);
getch();
closegraph();
}
```

OUTPUT:



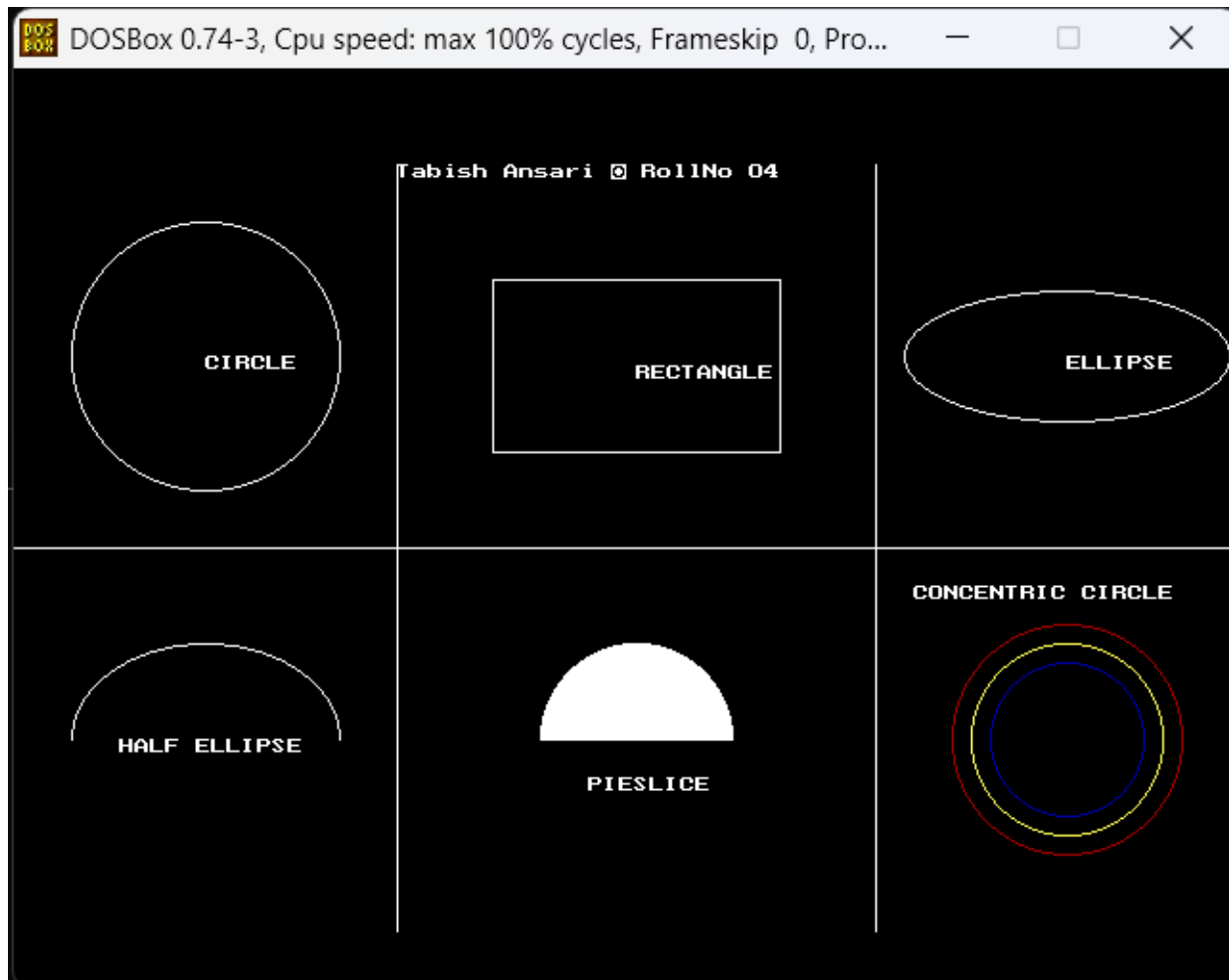
Experiment-2:

AIM: Divide your screen into region draw circle, rectangle, ellipse, half ellipse, square, & concentric circle in each region with appropriate message.

CODE:

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
void main(){
int gd = DETECT,gm,color;
initgraph(&gd,&gm,"C:\\\\TURBOC3\\\\BGI");
outtextxy(200,50,"Tabish Ansari \n RollNo 04");
line(0,250,850,250);
line(200,50,200,450);
line(450,50,450,450);
circle(100,150,70);
outtextxy(100,150,"CIRCLE");
rectangle(250,110,400,200);
outtextxy(325,155,"RECTANGLE");
ellipse(550,150,0,360,85,34);
outtextxy(550,150,"ELLIPSE");
ellipse(100,350,0,180,70,50);
outtextxy(55,350,"HALF ELLIPSE");
pieslice(325,350,0,180,50);
outtextxy(300,370,"PIESLICE");
outtextxy(470,270,"CONCENTRIC CIRCLE");
setcolor(BLUE);
circle(550,350,40);
setcolor(YELLOW);
circle(550,350,50);
setcolor(RED);
circle(550,350,60);
getch();
closegraph();
}
```

OUTPUT:



Experiment-2:

AIM: Draw a simple hut on screen.

CODE:

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
void main(){
int gd = DETECT,gm,color;
int i;
initgraph(&gd,&gm,"C:\\\\TURBOC3\\\\BGI");
outtextxy(200,370,"Tabish Ansari \n RollNo 04");
line(10,250,50,200);//one triangle tip
line(50,200,90,250);
line(10,250,90,250);
rectangle(10,250,90,350);
line(50,200,250,200);
line(250,200,300,250);
line(90,250,300,250);
rectangle(90,250,300,350);
//darwaza
rectangle(35,300,65,350);
setfillstyle(9,6);
floodfill(36,301,WHITE);
circle(58,325,3);//handlee
circle(50,230,10);
line(250,200,210,250);
rectangle(210,250,300,350);
//darwaza2.o
rectangle(240,300,270,350);
setfillstyle(9,6);
floodfill(241,301,WHITE);
circle(263,325,3); //hadle
circle(252,230,10);
//khidki
rectangle(125,270,175,300);
//---
line(125,270,145,280);//khidki ka mirror
line(145,280,145,290);
line(125,300,145,290);
//----
line(175,270,155,280);
line(155,280,155,290);
line(175,300,155,290);
```

```

//----coloring part
//triangle tip
setfillstyle(5,8);
floodfill(52,201,WHITE);
floodfill(11,251,WHITE);
floodfill(211,251,WHITE);
setfillstyle(2,1);
floodfill(91,251,WHITE);
setfillstyle(1,8);
floodfill(50,201,WHITE);
floodfill(250,201,WHITE);
setfillstyle(4,7);
floodfill(127,272,WHITE);
floodfill(174,273,WHITE);
//ghar complete
//mountais
setcolor(BROWN);
ellipse(100,160,0,180,100,150);
ellipse(280,160,0,153,100,150);
ellipse(460,160,0,153,100,150);
ellipse(640,160,0,153,100,150);
line(0,160,800,160);
setfillstyle(6,6);
floodfill(190,140,BROWN);
floodfill(300,140,BROWN);
floodfill(590,140,BROWN);
floodfill(450,140,BROWN);
setcolor(WHITE);
//KUWA
ellipse(450,250,0,360,80,25);
ellipse(450,260,0,180,70,15);
line(370,250,370,325);
line(530,250,530,325);
ellipse(450,325,180,360,80,25);
bar3d(370,245,375,200,10,5);
bar3d(515,250,520,200,10,5);
bar3d(372,205,513,212,10,5);
getch();
closegraph();
}

```

OUTPUT:



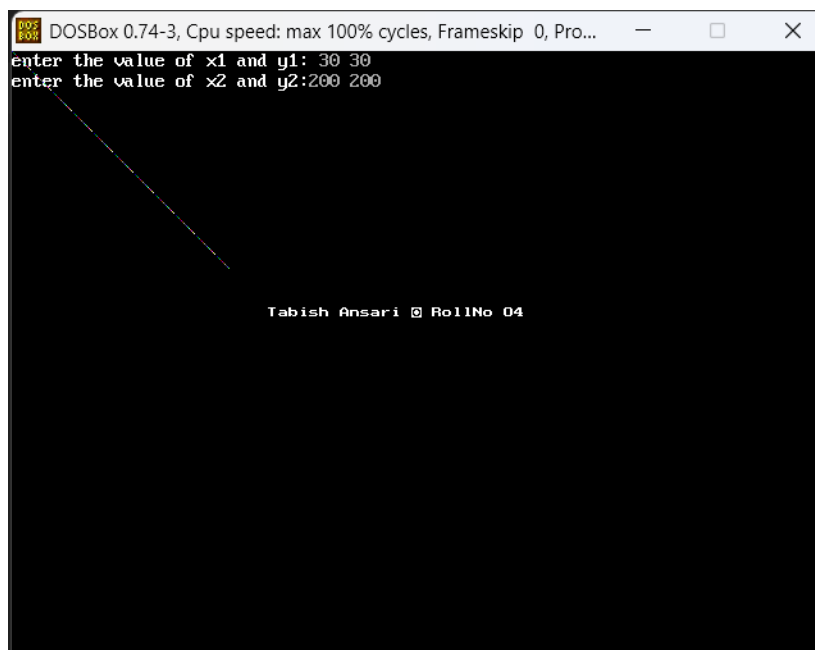
Experiment-3:

AIM: Develop the program for DDA line drawing algorithm.

CODE:

```
#include<graphics.h>
#include<stdio.h>
#include<math.h>
#include<dos.h>
void main(){
float x,y,x1,y1,x2,y2,dx,dy,step;
int i,gd=DETECT,gm;
initgraph(&gd,&gm,"c:\\TURBOC3\\BGI");
outtextxy(200,200,"Tabish Ansari \n RollNo 04");
printf("enter the value of x1 and y1: ");
scanf("%f%f",&x1,&y1);
printf("enter the value of x2 and y2:");
scanf("%f%f",&x2,&y2);
dx=abs(x2-x1);
dy=abs(y2-y1);
if(dx>=dy){ step=dx; }
else{ step=dy; }
dx=dx/step;
dy=dy/step;
i=1;
while(i<=step){
putpixel(x,y,i);
x=x+dx;
y=y+dy;
i=i+1;
delay(100); }
getch();
closegraph(); }
```

OUTPUT:



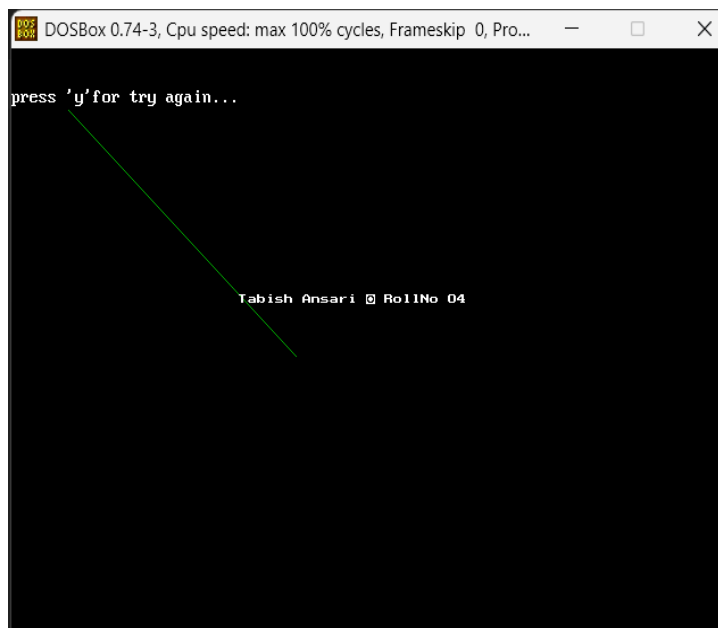
Experiment-4:

AIM: Develop program for bresenham line drawing algorithm.

CODE:

```
#include<graphics.h>
#include<stdio.h>
void main(){
int i,gd=DETECT,gm,x1,y1,x2,y2,dx,dy,pk;
do {
initgraph(&gd,&gm,"c:\\turbo3\\bgi");
printf("enter start coordinate:");
scanf("%d%d",&x1,&y1);
printf("enter the ending co ordinates:");
scanf("%d%d",&x2,&y2);
cleardevice();
dx=x2-x1;
dy=y2-y1;
pk=2*(dy-dx);
do{
putpixel(x1,y1,GREEN);
if(pk<0)
pk+=2*(dy-dx);
else {
++y1;
pk+=2*(dy-dx); }
++x1; }
while(x1<=x2);
printf("press 'y'for try again...");
outtextxy(200,200,"Tabish Ansari \n RollNo 04"); }
while(getch()!='y');
```

OUTPUT:



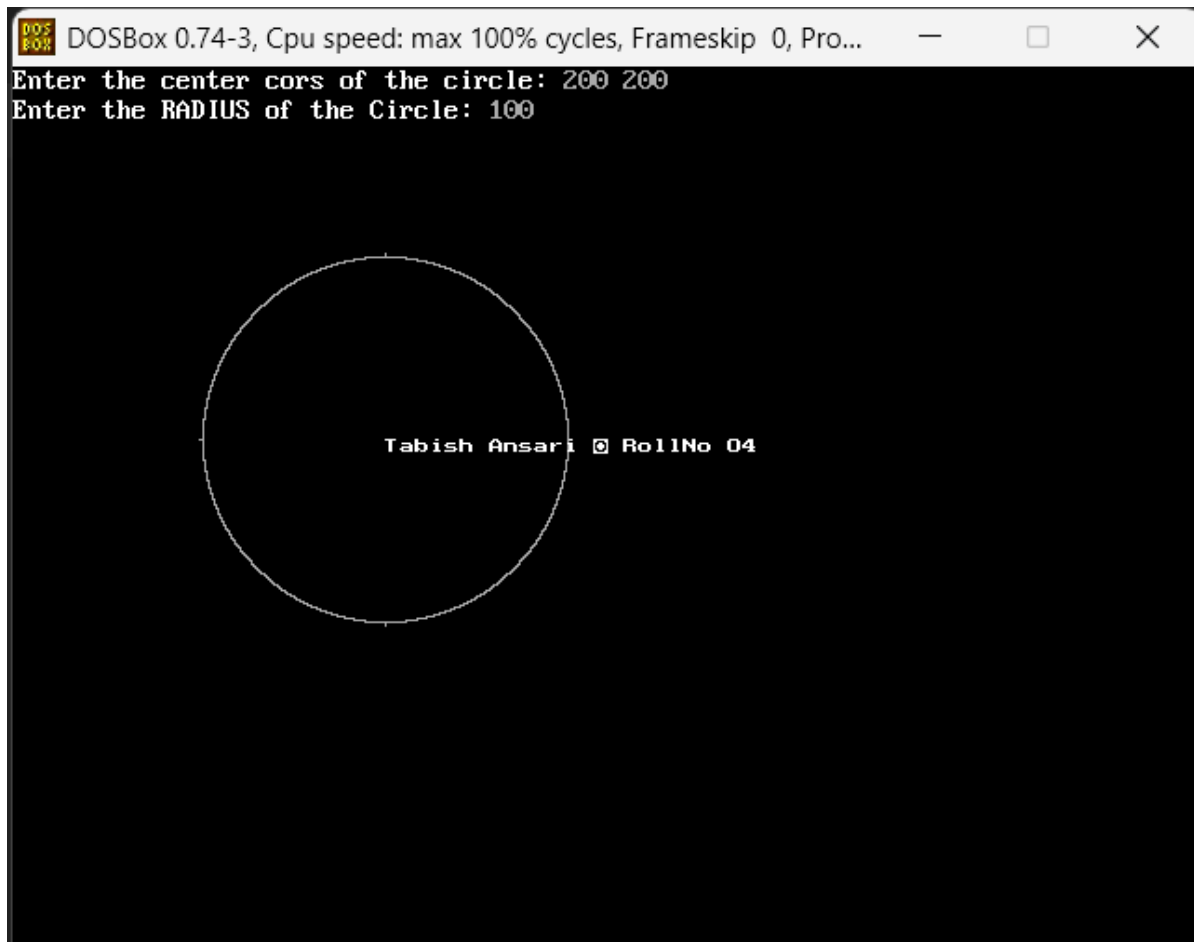
Experiment-5:

a) Develop the program for mid-point circle drawing algorithm.

Ans:

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
void drawcircle(int x0, int y0, int radius){
    int x=radius;
    int y = 0;
    int p;
    while(x>=y){
        putpixel(x0+x,y0+y,7);
        putpixel(x0+y,y0+x,7);
        putpixel(x0-x,y0+y,7);
        putpixel(x0-y,y0+x,7);
        putpixel(x0-x,y0-y,7);
        putpixel(x0-y,y0-x,7);
        putpixel(x0+x,y0-y,7);
        putpixel(x0+y,y0-x,7);
        if(p<=0){
            y=y+1;
            p=p+2*y+1;
        } else{
            x=x-1;
            p=p-2*x+1; }}}
void main(){
    int gd = DETECT,gm,color;
    int i,x,y,radi;
    initgraph(&gd,&gm,"C:\\\\TURBOC3\\\\BGI");
    outtextxy(300,300,"Fatima Momin");
    outtextxy(300,320,"Roll No: 66");
    printf("Enter the center cors of the circle: ");
    scanf("%d%d",&x,&y);
    printf("Enter the RADIUS of the Circle: ");
    scanf("%d",&radi);
    drawcircle(x,y,radi);
    getch();
    closegraph();
}
```

OUTPUT:



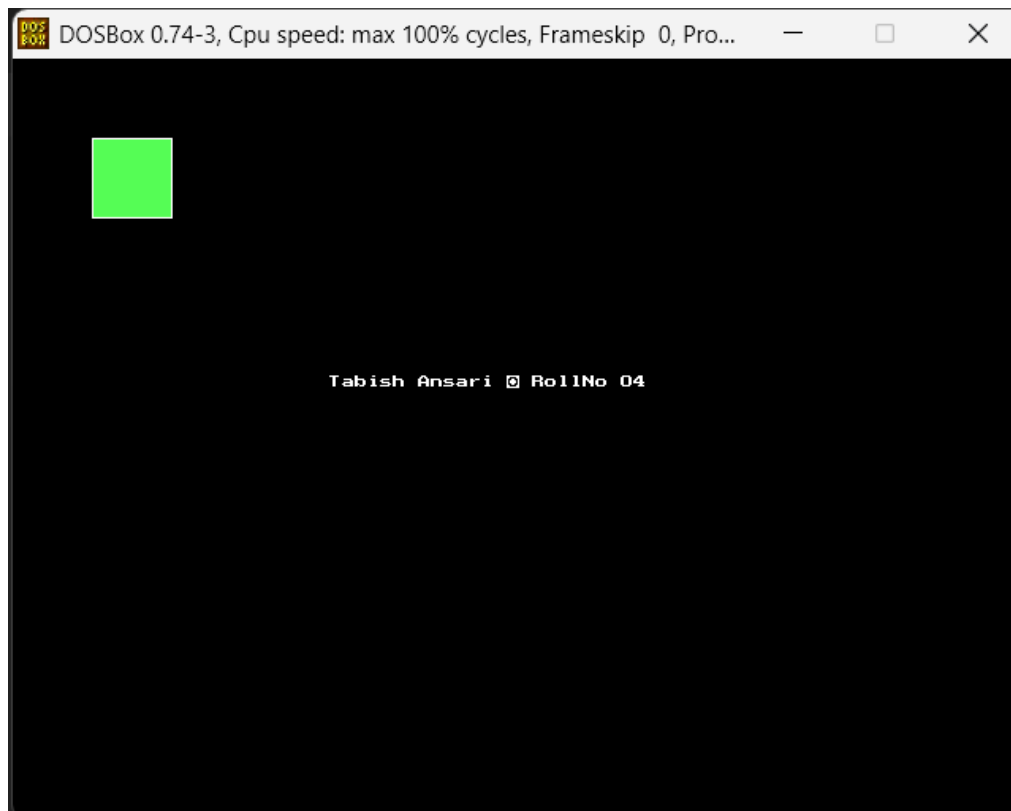
Experiment-6:

AIM: Write a program to fill a rectangle using Boundaryfill algorithm.

CODE:

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
void boundaryfill(int x, int y, int bordercolor, int fillcolor){
if(getpixel(x,y) != bordercolor && getpixel(x,y) != fillcolor){
putpixel(x,y,fillcolor);
boundaryfill(x+1,y,bordercolor,fillcolor);
boundaryfill(x,y+1,bordercolor,fillcolor);
boundaryfill(x-1,y,bordercolor,fillcolor);
boundaryfill(x,y-1,bordercolor,fillcolor);
}
}
void main(){
int gd = DETECT,gm,color;
initgraph(&gd,&gm,"C:\\\\TURBOC3\\\\BGI");
rectangle(50,50,100,100);
boundaryfill(55,55,15,10);
outtextxy(200,200,"Tabish Ansari \n RollNo 04");
getch();
closegraph();
}
```

OUTPUT:



Experiment-7:

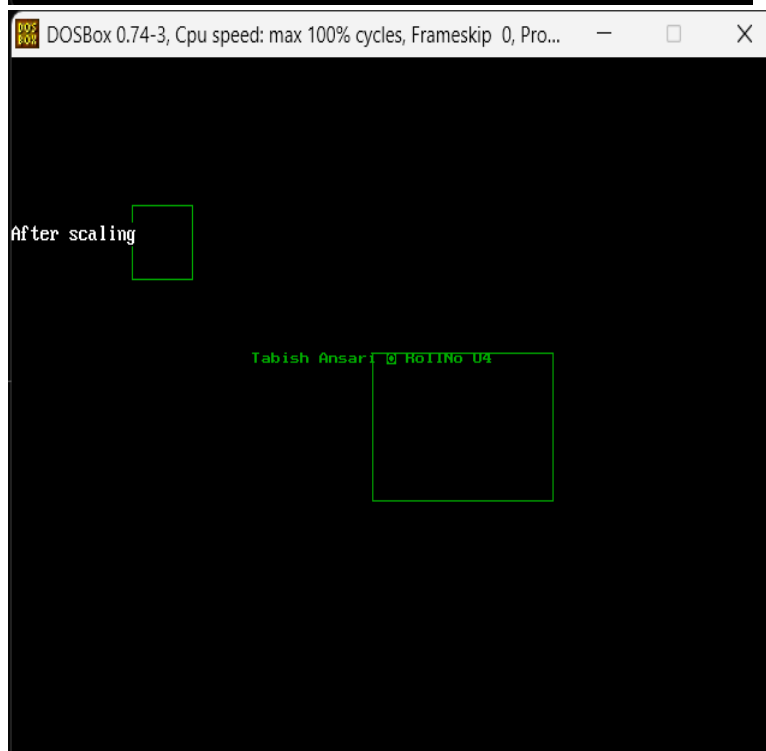
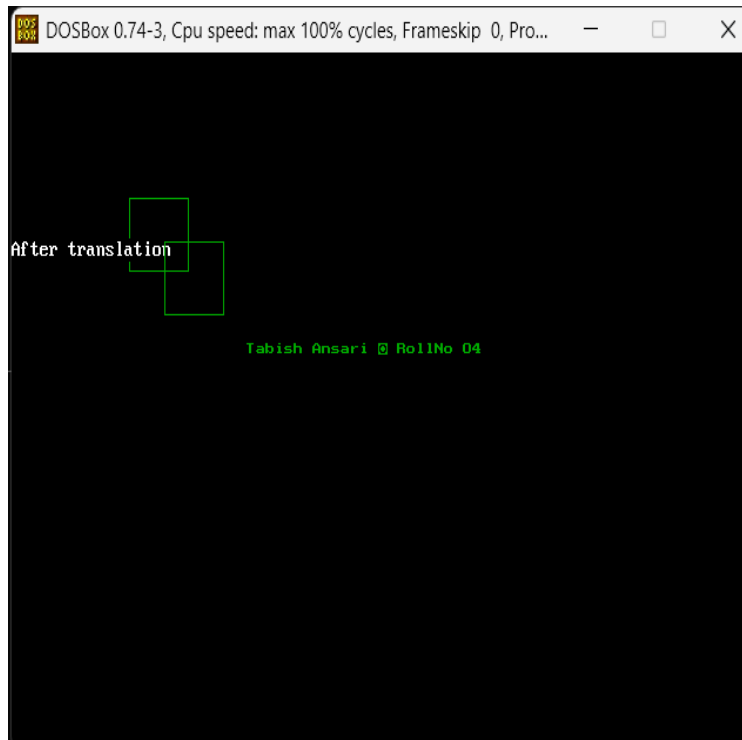
AIM: Write a program to implement 2D translation & scaling.

CODE:

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
#include<math.h>
void translate();
void scale();
void main(){
int ch;
int gd = DETECT,gm;
initgraph(&gd,&gm,"C:\\\\TURBOC3\\\\BGI");
setcolor(6);
outtextxy(100,88,"OBJ BEFORE");
rectangle(100,150,150,100);
printf("----MENU----");
printf("\\n1)Translation\\n2)Scale\\nEnter your choice");
scanf("%d",&ch);
cleardevice();
switch(ch)
{ case 1: translate();
break;
case 2: scale();
break;
default: printf("galat choice");
break; }
outtextxy(200,200,"Tabish Ansari \\n RollNo 04");
getch();
closegraph(); }
void translate(){
int tx,ty;
setcolor(2);
outtextxy(240,19,"Translation");
printf("\\nEnter tx: ");
scanf("%d",&tx);
printf("\\nEnter ty: ");
scanf("%d",&ty);
cleardevice();
rectangle(100,150,150,100);
printf("After translation");
rectangle(100+tx,150+ty,150+ty,100+ty); }
void scale(){
int sx,sy;
setcolor(2);
outtextxy(240,19,"Scaling--");
```

```
printf("\nEnter sx: ");
scanf("%d",&sx);
printf("Enter sy:");
scanf("%d",&sy);
cleardevice();
rectangle(100,150,150,100);
printf("After scaling");
rectangle(100*sx,150*sy,150*sx,100*sy); }
```

OUTPUT:



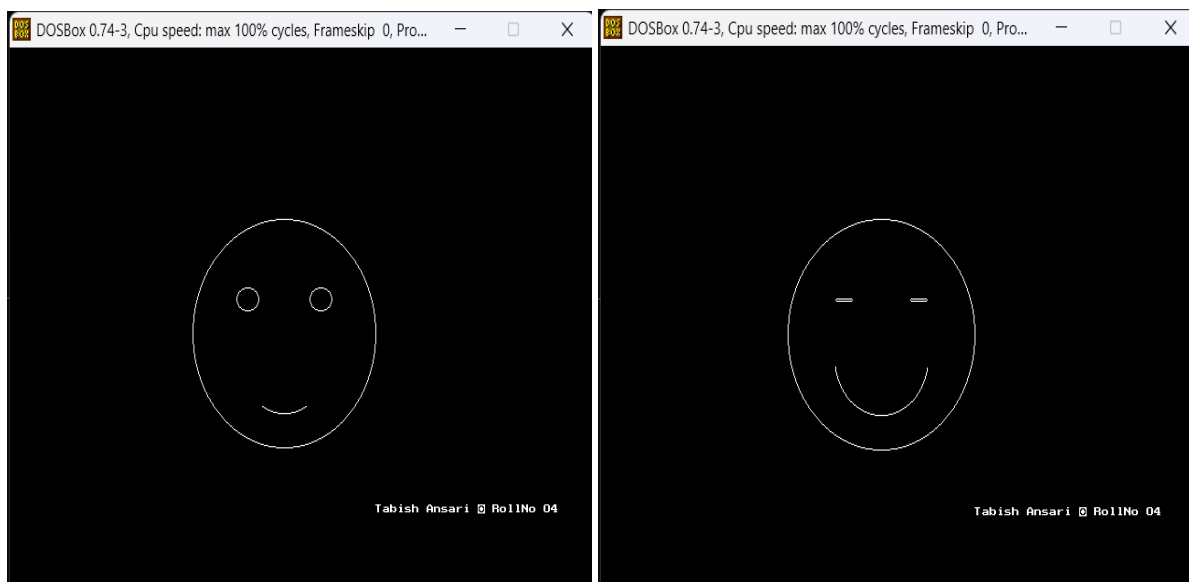
Experiment-8:

AIM: Perform Smiling face animation using graphics function.

CODE:

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
#include<math.h>
void main(){
int ch,i,j;
int gd = DETECT,gm;
initgraph(&gd,&gm,"C:\\\\TURBOC3\\\\BGI");
for(i=1;i<=12;i++){
circle(300,250,100);
ellipse(260,220,0,360,12,12-i);
ellipse(340,220,0,360,12,12-i);
for(j=1;j<=5;j++){
arc(300,270,250-j*i,290+j*i,50); }
delay(500);
cleardevice();
outtextxy(400,400,"Tabish Ansari \n RollNo 04");
}
getch();
closegraph(); }
```

OUTPUT:



Experiment-9:

AIM: Draw a moving car on screen.

CODE:

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
void main(){
int gd = DETECT,gm,color;
int i;
initgraph(&gd,&gm,"C:\\TURBOC3\\BGI");
for(i=0;i<500;i++){
outtextxy(200,20,"Tabish Ansari");
outtextxy(200,40,"Roll No: 04");
line(0,350,700,350);
circle(25+i,340,10);
circle(105+i,340,10);
line(5+i,330,125+i,330);
line(5+i,330,5+i,320);
line(125+i,330,125+i,320);
line(5+i,320,15+i,320);
line(5+i,320,125+i,320);
line(125+i,320,115+i,320);
line(15+i,320,50+i,290);
line(115+i,320,80+i,290);
line(50+i,290,80+i,290);
line(65+i,290,65+i,320);
delay(10);
cleardevice(); }
getch();
closegraph(); }
```

OUTPUT:



Mini-Project:

AIM: Draw any Cartoon Character using C graphics.

CODE:

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
void main(){
int gd = DETECT,gm,color;
int i;
initgraph(&gd,&gm,"C:\\\\TURBOC3\\\\BGI");
arc(300,150,0,110,70);
ellipse(410,165,140,170,50,25);
line(361,160,361,130);
ellipse(300,151,20,130,65,65);
ellipse(320,170,0,30,60,40);
ellipse(320,123,130,180,70,50);
line(265,100,280,85);
line(280,85,295,100);
line(295,100,280,95);
line(280,95,265,100);
line(300,100,315,85);
line(315,85,330,100);
line(330,100,315,95);
line(315,95,300,100);
arc(280,115,10,170,15);
arc(315,115,10,170,15);
circle(285,125,14);
setfillstyle(SOLID_FILL,WHITE);
circle(285,125,5);
floodfill(285,125,WHITE);
circle(325,125,14);
circle(325,125,5);
floodfill(325,125,WHITE);
ellipse(250,158,90,270,37,35);
line(250,193,325,185);
line(325,185,390,167);
line(390,167,395,175);
line(395,175,350,210);
line(214,165,204,157);
line(204,157,195,165);
line(195,165,237,210);
ellipse(260,175,0,360,10,15);
line(350,210,354,310);
line(237,210,233,310);
line(233,310,285,312);
line(354,310,302,312);
```



```
line(285,312,293,300);
line(302,312,290,295);
//pair
line(253,310,250,350);
line(265,310,262,350);
//pair/jute
line(262,350,250,350);
line(250,350,240,355);
line(240,355,240,360);
line(240,360,265,360);
line(265,360,262,350);
line(252,335,263,335);
//pair 2.o
line(327,310,330,350);
line(339,310,342,350);
line(330,335,342,335);
line(330,350,342,350);
//pair2.o/jute
line(330,350,320,355);
line(320,355,320,360);
line(320,360,345,360);
line(345,360,342,350);
//pair done
line(353,270,233,270);
//ungli leftstart = 390,167
line(391,170,395,165);
line(395,165,393,158);
line(393,158,405,160);
line(405,160,398,162);
line(398,162,400,166);
line(397,158,402,145);
line(402,145,402,158);
line(402,158,415,150);
line(415,150,408,160);
line(393,174,400,169);
line(400,169,404,173);
line(404,173,412,165);
line(412,165,408,160);
//ungli rightstart =204,157
line(202,160,197,155);
line(197,155,199,147);
line(199,147,189,146);
line(189,146,193,150);
line(193,150,192,155);
line(197,147,194,132);
line(194,132,192,145);
line(192,145,185,135);
line(185,135,186,145);
line(186,145,182,150);
```

```
line(182,150,187,160);
line(197,164,192,159);
line(192,159,187,160);
setcolor(RED);
settextstyle(10,0,2);
outtextxy(230,20,"SHINCHAN");
setcolor(YELLOW);
settextstyle(7,0,3);
outtextxy(420,160,"-Made By");
outtextxy(420,190,"Tabish Ansari");
outtextxy(420,220,"Roll.No: 04 :)");
getch();
closegraph();
}
```

OUTPUT:

