SIR C.R.R COLLEGE OF ENGINEERING, ELURU.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

GRAPHICS AND MULTIMEDIA LAB MANUAL

ACADEMIC YEAR: 2016-2017

CLASS : 4/4 B.Tech CSE I SEMESTER

SUBJECT : Graphics & Multimedia lab CSE 4.1.7

FACULTY: K. N. Madhavi Latha

J.S.V.Gopala Krishna

S.M.B.Chowdary

LIST OF PROGRAMS

GRAPHICS PROGRAMS using graphics package in C

MULTIMEDIA PROGRAMS

Programs

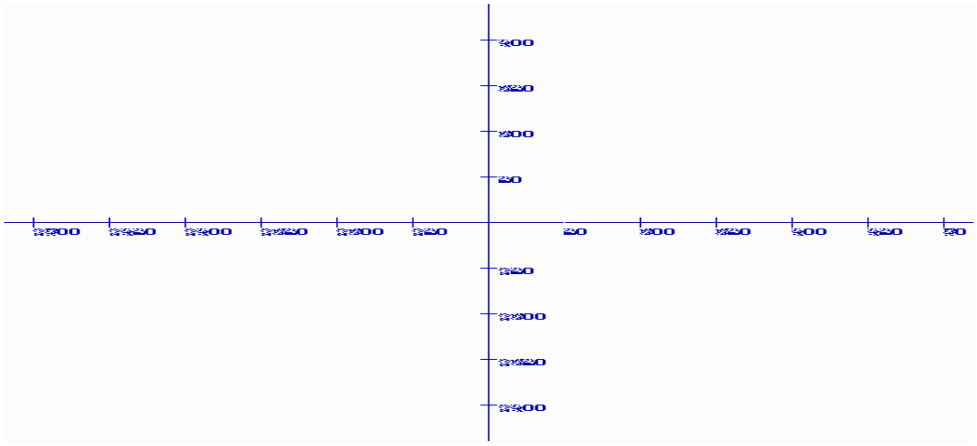
program 1: Write a program to generate coordinate axis.

```
#include<stdio.h>
#include<graphics.h>
#include<conio.h>
#include<dos.h>
main()
{
        int gd=DETECT,gm;
        int xm,ym,i,j=0;
        char str[20];
        initgraph(&gd,&gm," ");
        setbkcolor(15);
        xm=getmaxx();
        ym=getmaxy();
        for(i=0;i<xm;i++)
                plot(i,ym/2,1);
        sleep(.1);
        for(i=0;i<ym;i++)
                plot(xm/2,i,1);
        sleep(.1);
        for(i=(xm/2)+50,j=50;i<xm;i=i+50,j=j+50)
        {
                line(i,(ym/2)-5,i,(ym/2)+5);
```

```
        itoa(j,str,10);
        setcolor(1);
        outtextxy(i,(ym/2)+7,str);
}
sleep(.1);
for(i=(ym/2)+50,j=50;i<ym;i=i+50,j=j+50)
{
        line(xm/2-5,i,xm/2+5,i);
        itoa(-j,str,10);
        setcolor(1);
        outtextxy(xm/2+7,i,str);
}
sleep(.1);
for(i=ym/2-50,j=50;i>=0;i=i-50,j=j+50)
{
        line(xm/2-5,i,xm/2+5,i);
        itoa(j,str,10);
        setcolor(1);
        outtextxy(xm/2+7,i,str);
}
sleep(.1);
for(i=xm/2-50,j=50;i>=0;i=i-50,j=j+50)
{
        line(i,ym/2-5,i,ym/2+5);
        itoa(-j,str,10);
```

```
                setcolor(1);
                outtextxy(i,ym/2+7,str);
        }
        sleep(.5);
return (0);
}       plot(int x,int y,int c)
        {
                union REGS i,o;
                i.h.ah=12;
                i.h.al=c;
                i.x.cx=x;
                i.x.dx=y;
                i.h.bh=0;
                int86(16,&i,&o);
                return(0);
        }
```

OUTPUT:



Program 2: BRESENHAMS LINE GENERATION ALGORITHM

Aim : Write a program to implement Bresenham's line generation algorithm.

/*BRESENHAM'S LINE GENERATION */

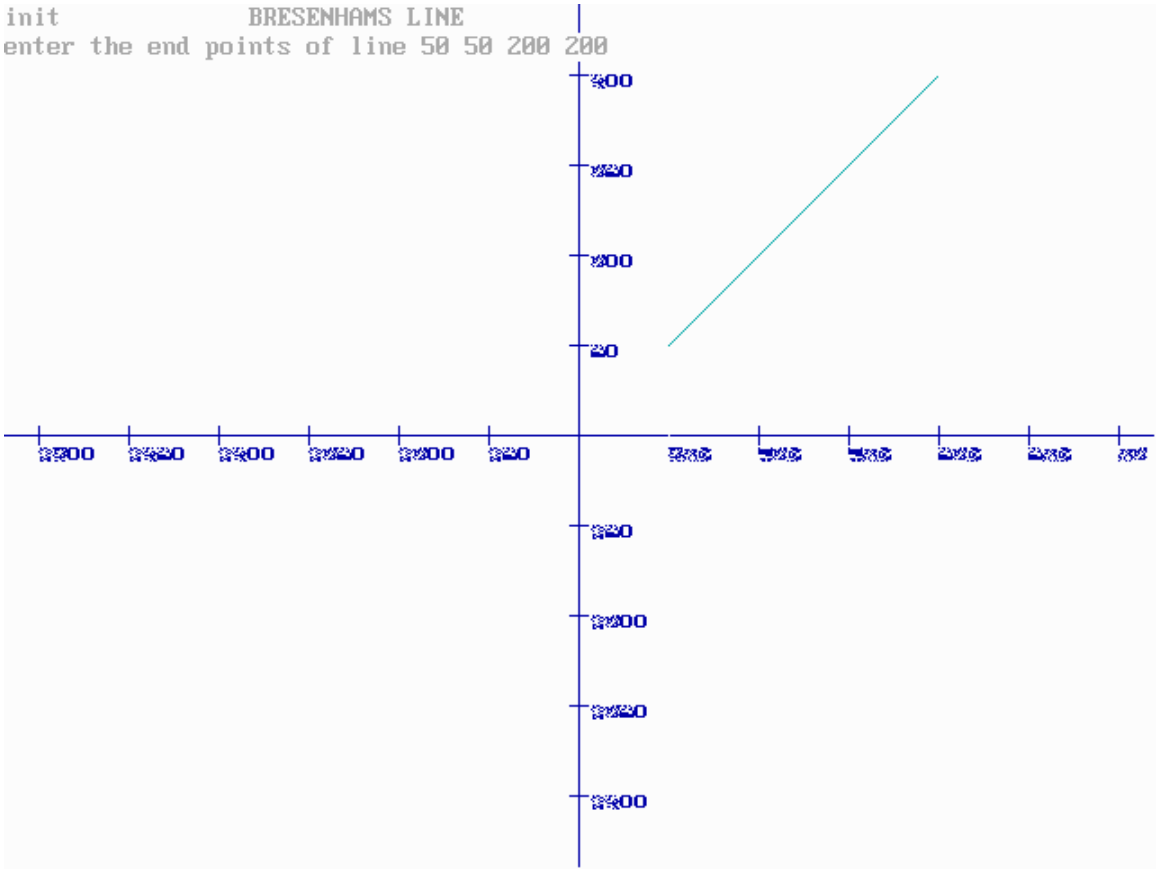
```
#include<stdio.h>
#include<graphics.h>
#include<math.h>
#include<conio.h>
#include<dos.h>
#include "c:\teja\axis.c"
main()
{
        int gd=DETECT,gm;
        int xm,ym,i,x1,x2,y1,y2,x,y,s1,s2,e,flag,dx,dy,temp;
        initgraph(&gd,&gm," ");
        setbkcolor(15);
        xm=getmaxx();
        ym=getmaxy();
        axis();
        printf("\n BRESENHAMS LINE\n ");
        printf("\nENTER THE END POINTS OF LINE:");
        scanf("%d%d%d%d",&x1,&y1,&x2,&y2);
        x1=xm/2+x1;
```

```
 x2=xm/2+x2;
 y1=ym/2-y1;
 y2=ym/2-y2;
 x=x1;
 y=y1;
dx=abs(x2-x1);
 dy=abs(y2-y1);
 if((x2-x1)<0)
        s1=-1;
 else
 {
        if((x2-x1)>0)
                s1=1;
        else
                s1=0;
 }
 if((y2-y1)<0)
        s2=-1;
 else
 {
        if((y2-y1)>0)
                s2=1;
        else
                s2=0;
 }
```

```
if(dy>dx)
{
        temp=dx;
        dx=dy;
        dy=temp;
        flag=1;
}
else
        flag=0;
e=2*dy-dx;
for(i=0;i<dx;i++)
{               putpixel(x,y,3);
        while(e>=0)
        {
                if(flag==1)
                        x=x+s1;
                else
                        y=y+s2;
        e=e-2*dx;
        }
        if(flag==1)
                y=y+s2;
        else
                x=x+s1;
e=e+2*dy;
```

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

OUTPUT:



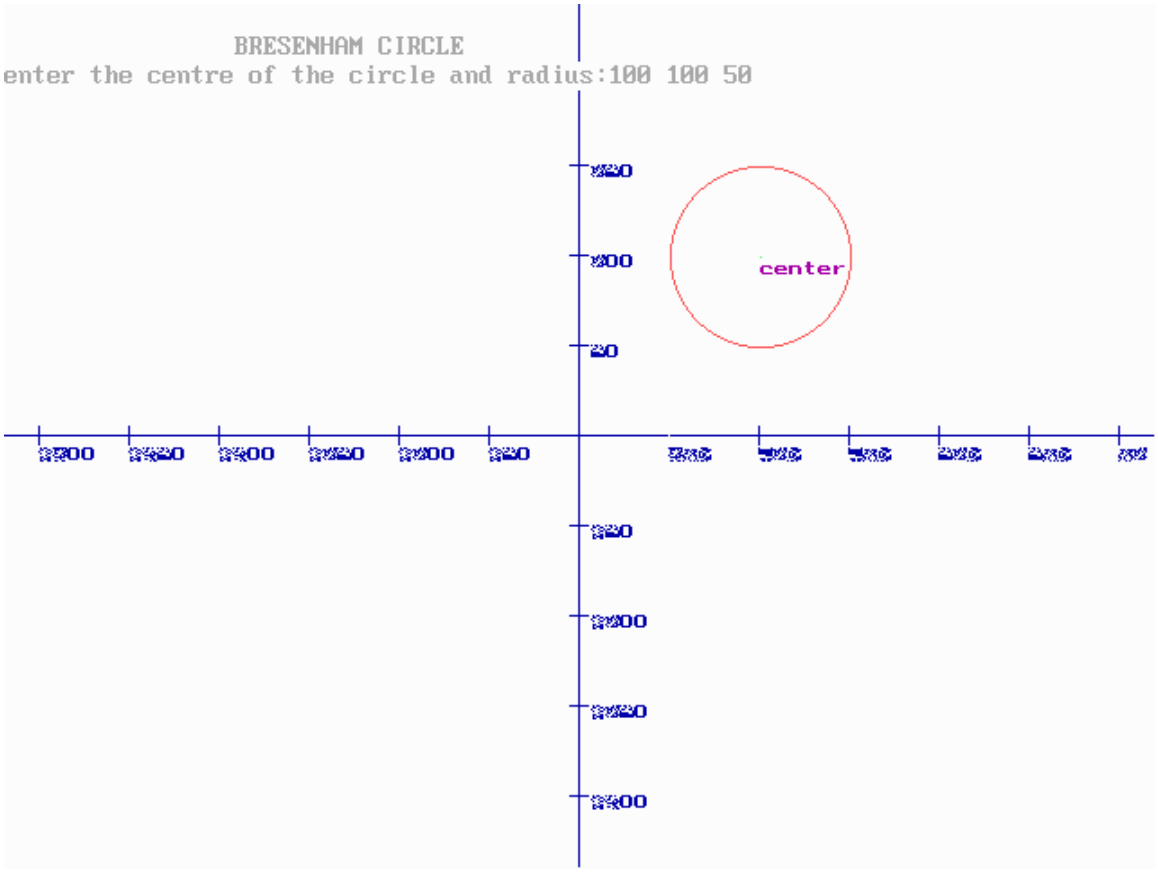
Program 3: Program to draw a Circle using Midpoint Algorithm

/* CIRCLE GENERATION*/

```
#include<stdio.h>

#include<conio.h>

#include<graphics.h>

#include "c:\teja\axis.c"

void circleMidpoint(int,int,int);
void drawCircle(int,int,int,int);
void main()
{
int xc,yc,r;
int gd=DETECT ,gm;
                axis();

                printf("\n\tMid point  circle");

                printf("\nenter the center of the circle and radius");

                scanf("%d%d%d",&xc,&yc,&r);

                circleMidpoint(xc,yc,r);

                getch();

                }

void circleMidpoint(int xc,int yc,int r)
{

        int x=0,y=r;

        int p=1-r;

        while(x<y)

        {
```

```
            drawCircle(xc,yc,x,y);
             x=x+1;
             if(p<0)
                    p=p+2*x+1;
             else
             {
                    y=y-1;
                    p=p+2*(x-y)+1;
             }
             drawCircle(xc,yc,x,y);
             delay(60);
      }
}
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);
}
```

OUTPUT:

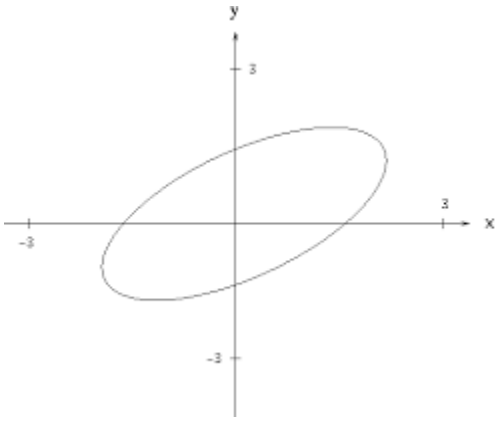

BRESENHAM CIRCLE
enter the centre of the circle and radius:100 100 50
center

Program 4: Program to draw an Ellipse using Midpoint Algorithm

```
#include <stdio.h>
#include <conio.h>
#include <graphics.h>
#include <math.h>
#include <dos.h>
void ellipseMidpoint(float,float,float,float);
void drawEllipse(float,float,float,float);
void main()
{
int xc,yc,rx,ry;
int gd=DETECT ,gm;
initgraph(&gd,&gm," ");
printf("Enter the center coordinates of the ellipse");
scanf("%f %f",&xc,&yc);
printf("Enter the x Radius of the ellipse");
scanf("%f",&rx);
printf("Enter the y Radius of the ellipse");
scanf("%f",&ry);
ellipseMidpoint(xc,yc,rx,ry);
getch();
}
void ellipseMidpoint(float xc,float yc,float rx,float ry)
{
float rxsq=rx*ry;
float rysq=ry*ry;
float x=0,y=ry,p;
float px=0, py=2*rxsq*ry;
drawEllipse(xc,yc,x,y);
//region 1
p=rysq-(rxsq*ry)+(0.25*rxsq);
while(px<py)
{
x++;
px=px+2*rysq;
if(p<0)
p=p+rysq+px;
else
```

```
{
y--;
py=py-2*rxsq;
p=p+rysq+px-py;
}
drawEllipse(xc,yc,x,y);
delay(30);
}
//region 2
P=rysq*(x+0.5)*(x+0.5)+rxsq*(y-1)*(y-1)-rxsq*rysq;
While(y>0)
{
y--;
py=py-2*rxsq;
if(p>0)
p=p+rxsq-py;
else
{
x++;
px=px+2*rysq;
p=p+rxsq-py+px;
}
drawEllipse(xc,yc,x,y);
delay(30);
}
}

Void drawEllipse(float xc, float yc, float x, float 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);
  }
```

OUTPUT:



Program 5: Write a program to create various types of Texts & Fonts.

```
/*FONT*/
#include<graphics.h>
#include<conio.h>
#include<stdio.h>
int font[8][8],size;
main()
{
        int gd=DETECT,gm,i,j,k;
        char ch,s[40];
        initgraph(&gd,&gm,"");
        setbkcolor(15);
        printf("Enter any character:");
        ch=toupper(getchar());
        printf("\Enter ur font size(or) else enter 0:");
        scanf("%d",&size);
        switch(ch)
        {
                case 'A':
                        for(i=1;i<7;i++)
                                font[i][2]=1;
                        for(i=1;i<7;i++)
                        font[i][4]=1;
```

```
        font[1][3]=font[4][3]=1;
        display();
        break;
case 'B':
        for(i=0;i<7;i++)
                font[i][2]=1;
        font[0][3]=font[3][3]=font[6][3]=1;
        font[1][4]=font[2][4]=font[4][4]=1;
        font[5][4]=1;
        display();
        break;
case 'C':
        for(i=2;i<6;i++)
                font[i][2]=1;
        font[1][3]=font[1][4]=1;
        font[2][5]=font[5][5]=1;
        font[6][3]=font[6][4]=1;
        display();
        break;
case 'D':
        for(i=0;i<7;i++)
                font[i][2]=1;
        font[0][3]=font[6][3]=1;
        for(i=1;i<6;i++)
                font[i][4]=1;
```

```
            display();
            break;
    case 'E':
            for(i=0;i<7;i++)
                    font[i][2]=1;
            font[0][3]=font[0][4]=1;
            font[3][3]=1;
            font[6][3]=font[6][4]=1;
            display();
            break;
    case 'F':
            for(i=0;i<7;i++)
                    font[i][2]=1;
                    font[0][3]=font[0][4]=1;
            font[3][3]=1;
            display();
            break;
    case 'G':
            for(i=2;i<6;i++)
                    font[1][i]=1;
            for(i=2;i<7;i++)
                    font[i][1]=1;
            for(i=2;i<6;i++)
                    font[7][i]=1;
            for(i=6;i>3;i--)
```

```
                font[i][5]=1;
        font[4][3]=font[4][4]=1;
        display();
        break;
case 'H':
        for(i=2;i<7;i++)
                font[i][2]=1;
        for(i=2;i<7;i++)
                font[i][4]=1;
        font[4][3]=1;
        display();
        break;
case 'I':
        for(i=1;i<7;i++)
                font[i][3]=1;
        font[1][2]=font[1][4]=1;
        font[6][2]=font[6][4]=1;
        display();
        break;
case 'J':
        for(i=1;i<7;i++)
                font[i][3]=1;
       font[1][2]=font[1][4]=1;
       font[6][2]=1;
       font[5][1]=1;
```

```
            display();
            break;
    case 'K':
            for(i=0;i<7;i++)
                    font[i][2]=1;
            font[3][3]=1;
            font[2][4]=font[4][4]=1;
            font[1][5]=font[5][5]=1;
            font[0][6]=font[6][6]=1;
            display();
            break;
    case 'L':
            for(i=1;i<7;i++)
                    font[i][2]=1;
            font[6][3]=font[6][4]=1;
            display();
            break;
    case 'M':
            for(i=1;i<7;i++)
                    font[i][1]=1;
            for(i=1;i<7;i++)
                    font[i][5]=1;
            font[2][2]=font[3][3]=1;
            font[2][4]=1;
            display();
```

```
            break;
case 'N':
            for(i=1;i<7;i++)
                  font[i][2]=1;
            for(i=1;i<7;i++)
                  font[i][6]=1;
       //   font[2][2]=font[3][3]=1;
            font[2][3]=font[3][4]=1;
           font[4][5]=1;
           display();
           break;
case 'O':
            for(i=1;i<6;i++)
                  font[i][2]=1;
            for(i=1;i<6;i++)
                  font[i][5]=1;
            font[0][3]=font[0][4]=1;
            font[6][3]=font[6][4]=1;
            display();
            break;
case 'P':
            for(i=1;i<7;i++)
                  font[i][2]=1;
            font[0][3]=font[0][4]=1;
            font[2][3]=font[2][4]=1;
```

```
        font[1][5]=1;
        display();
        break;
case 'Q':
        for(i=1;i<6;i++)
                font[i][2]=1;
        for(i=1;i<6;i++)
                font[i][5]=1;
        font[0][3]=font[0][4]=1;
        font[6][3]=font[6][4]=1;
        font[3][3]=font[3][4]=1;
        font[4][6]=1;
        display();
        break;
case 'R':
        for(i=1;i<7;i++)
                font[i][2]=1;
        font[0][3]=font[0][4]=font[0][5]=1;
        font[1][6]=font[2][5]=font[3][4]=1;
        font[4][4]=font[5][5]=font[6][6]=1;
        display();
        break;
case 'S':
        font[0][4]=font[0][5]=font[0][6]=1;
        font[1][3]=font[2][4]=1;
```

```
        font[3][5]=1;
        font[4][5]=font[5][5]=1;
        font[6][2]=font[6][3]=font[6][4]=1;
        display();
        break;
case 'T':
        for(i=1;i<7;i++)
                font[i][3]=1;
        font[1][2]=font[1][4]=1;
        font[1][1]=font[1][5]=1;
        display();
        break;
case 'U':
         for(i=1;i<7;i++)
                font[i][2]=1;
         for(i=1;i<7;i++)
                font[i][5]=1;
         font[6][4]=font[6][3]=1;
         display();
         break;
case 'V':
        font[2][1]=font[3][2]=font[4][3]=1;
        font[5][4]=font[4][5]=font[3][6]=font[3][2]=1;
        font[2][7]=1;
        display();
```

```
            break;
case 'W':
            for(i=1;i<7;i++)
                  font[i][2]=1;
            for(i=1;i<7;i++)
                  font[i][6]=1;
            font[4][3]=font[3][4]=font[4][5]=1;
            display();
            break;
case 'X':
            font[1][1]=font[2][2]=1;
            font[1][5]=font[2][4]=1;
            font[3][3]=1;
            font[5][1]=font[4][2]=1;
            font[5][5]=font[4][4]=1;
            display();
            break;
case 'Y':
            font[1][1]=font[2][2]=1;
            font[1][5]=font[2][4]=1;
            font[3][3]=1;
            font[4][3]=font[5][3]=1;
            display();
            break;
case 'Z':
```

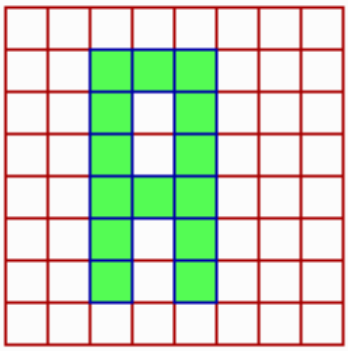
```
                for(i=2;i<6;i++)
                        font[1][i]=1;
                font[2][5]=1;
                font[3][4]=font[4][3]=1;
                for(i=2;i<6;i++)
                        font[5][i]=1;
                display();
                break;
        }
        getch();
        closegraph();
        restorecrtmode();
}
display()
{
        int i,j,k=1,n;
        if(size>0)
                n=size;
        else
                n=1;
        for(;n<=25;n++)
        {
                cleardevice();
                setcolor(4);
                outtextxy(10,10,"This is present font size:");
```

```
            printf("%d",n);
            rectangle(100,100,100+8*n,100+8*n);
            k=0;
            while(k<9)
            {
                  line(100,100+k*(8*n/8),100+8*n,100+k*(8*n/8));
                  line(100+k*(8*n/8),100,100+k*(8*n/8),100+8*n);
                  k++;
            }
            setcolor(1);
            for(i=0;i<8;i++)
            {
                  for(j=0;j<8;j++)
                  if(font[i][j]==1)
                  {
                        setfillstyle(1,10);

      bar3d(100+j*(8*n/8),100+i*(8*n/8),100+(j+1)*(8*n/8),100+(i+1)*(8*n/8),0
,1);
                  }
            }
            if(size>0)
                  break;
      }
}
```

OUTPUT:

```
this is present font size
```

15



Program 6:**To implement polygon filling algorithms**.

Polygon Filling :

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

main()
{

int n,i,j,k,gd,gm,dy,dx;
int x,y,temp;
int a[20][2],xi[20];
float slope[20];


clrscr();
printf("\n\n\tEnter the no. of edges of polygon : ");
scanf("%d",&n);
printf("\n\n\tEnter the cordinates of polygon :\n\n\n ");

for(i=0;i<n;i++)
{
printf("\tX%d Y%d : ",i,i);
scanf("%d %d",&a[i][0],&a[i][1]);
}

a[n][0]=a[0][0];
a[n][1]=a[0][1];


detectgraph(&gd,&gm);
initgraph(&gd,&gm,"c:\\tc\\bgi");


/*- draw polygon -*/

for(i=0;i<n;i++)
{
```

```
line(a[i][0],a[i][1],a[i+1][0],a[i+1][1]);
}

getch();


for(i=0;i<n;i++)
{
dy=a[i+1][1]-a[i][1];
dx=a[i+1][0]-a[i][0];

if(dy==0) slope[i]=1.0;
if(dx==0) slope[i]=0.0;

if((dy!=0)&&(dx!=0)) /*- calculate inverse slope -*/
{
slope[i]=(float) dx/dy;
}
}

for(y=0;y< 480;y++)
{
k=0;
for(i=0;i<n;i++)
{

if( ((a[i][1]<=y)&&(a[i+1][1]>y))||
((a[i][1]>y)&&(a[i+1][1]<=y)))
{
xi[k]=(int)(a[i][0]+slope[i]*(y-a[i][1]));
k++;
}
}

for(j=0;j<k-1;j++) /*- Arrange x-intersections in order -*/
for(i=0;i<k-1;i++)
{
if(xi[i]>xi[i+1])
{
temp=xi[i];
```

```
xi[i]=xi[i+1];
xi[i+1]=temp;
}
}

setcolor(35);
for(i=0;i<k;i+=2)
{
line(xi[i],y,xi[i+1]+1,y);
getch();
}

}

}
```

Output:

Enter no of edges of polygon

3

Enter the Co-ordinates of polygon

X0 y0:100 200

X1 Y1:200 300

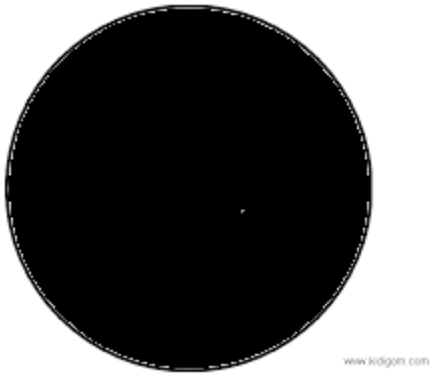
X2 Y2:300 400

Seed filling Algorithm:

```
#include<stdio.h>
#include<graphics.h>
#include<conio.h>
viod boundary_fill(int  sx, int  sy,int  fill _color, int  boundary_color)
viod main()
{
int gd=detect,gm,x,y,r;
initgraph(&gd,&gm,x,y,r);
setcolor(white);
putpixel(310,230,red)
printf("enter the x,y radius: \n");
scanf("%d%d%d",&x&y&r);
circle(x,y,r);
boundary_ fill(x,y, light green,white);
getch();
}


void boundary _fill(int sx,intsy,int  fill_color, int boundary_color)
{
if(getpixel(sx,sy)!=boundary_color && getpixel(sx,sy)!fill_color);
boundary_fill((sx+1),sy, fill_color, boundary_color);
boundary_fill((sx-1), sy, fill _color, boundary_color);
boundary_fill(sx, (sy+1),fill_color, boundary_color);
```

```
boundary_fill(sx, (sy-1), fill_color, boundary_color);
}
}
```

OUTPUT:



Program 7: Write a program for 2D transformations like scaling,rotating and transforming a polygon about an axis.

/*2D-TRANSFORMATIONS*/

```
#include<stdlib.h>
#include<math.h>
#include<conio.h>
#include<stdio.h>
#include"C:\teja\axis.c"
void rotate();
void scale();
void transulate();
int x[10],y[10],n,i;
void main()
{
        int ch;
        int gd=DETECT,gm;
        initgraph(&gd,&gm,"");
        while(1)
        {
                printf("\n2D Transformations");
                printf("\n1.Scaling\n2.Rotation\n3.Transulation\n4.Exit");
                printf("\nEnter your choice:");
                scanf("%d",&ch);
                if(ch!=4)
```

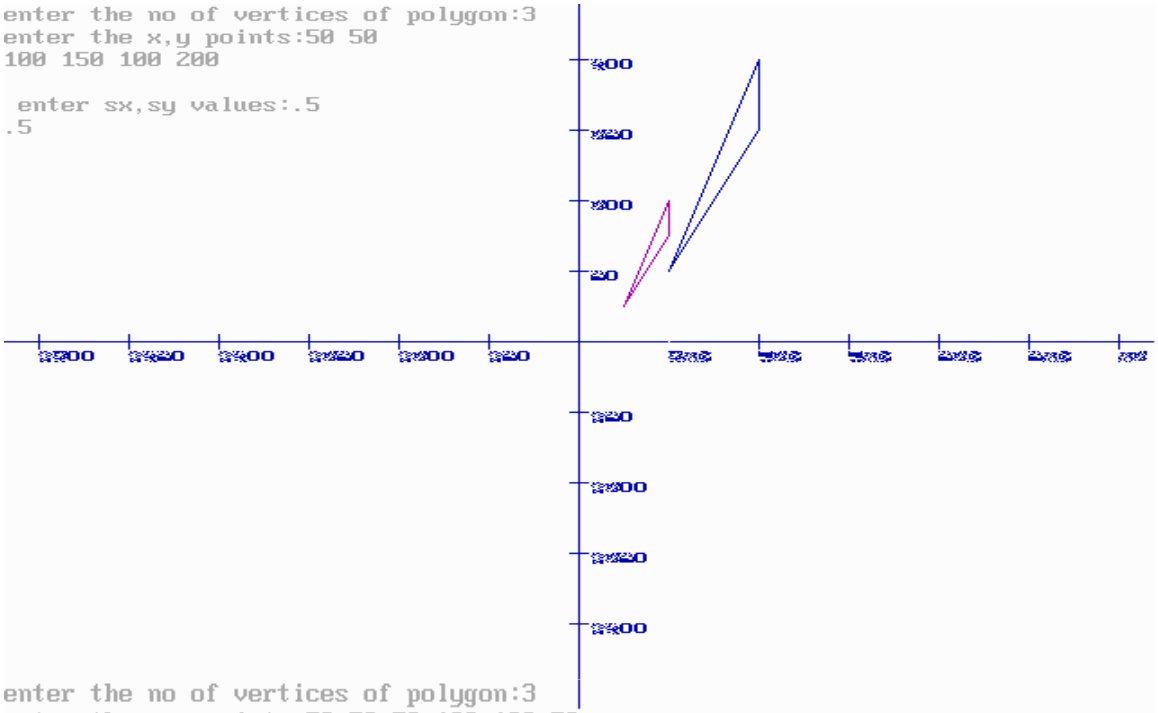
```
{
        cleardevice();
        axis();
        gotoxy(0,0);
        printf("Enter the no.of vertices of polygon:\n");
        scanf("%d",&n);
        printf("Enter the x,y points:");
        for(i=0;i<n;i++)
                scanf("%d%d",&x[i],&y[i]);
        for(i=0;i<n-1;i++)
            line(x[i]+319,239-y[i],319+x[i+1],239-y[i+1]);

        line(x[i]+319,239-y[i],319+x[0],239-y[0]);
}
switch(ch)
{
        case 1:
                scale();
                break;
        case 2:
                rotate();
                break;
        case 3:
                transulate();
                break;
```

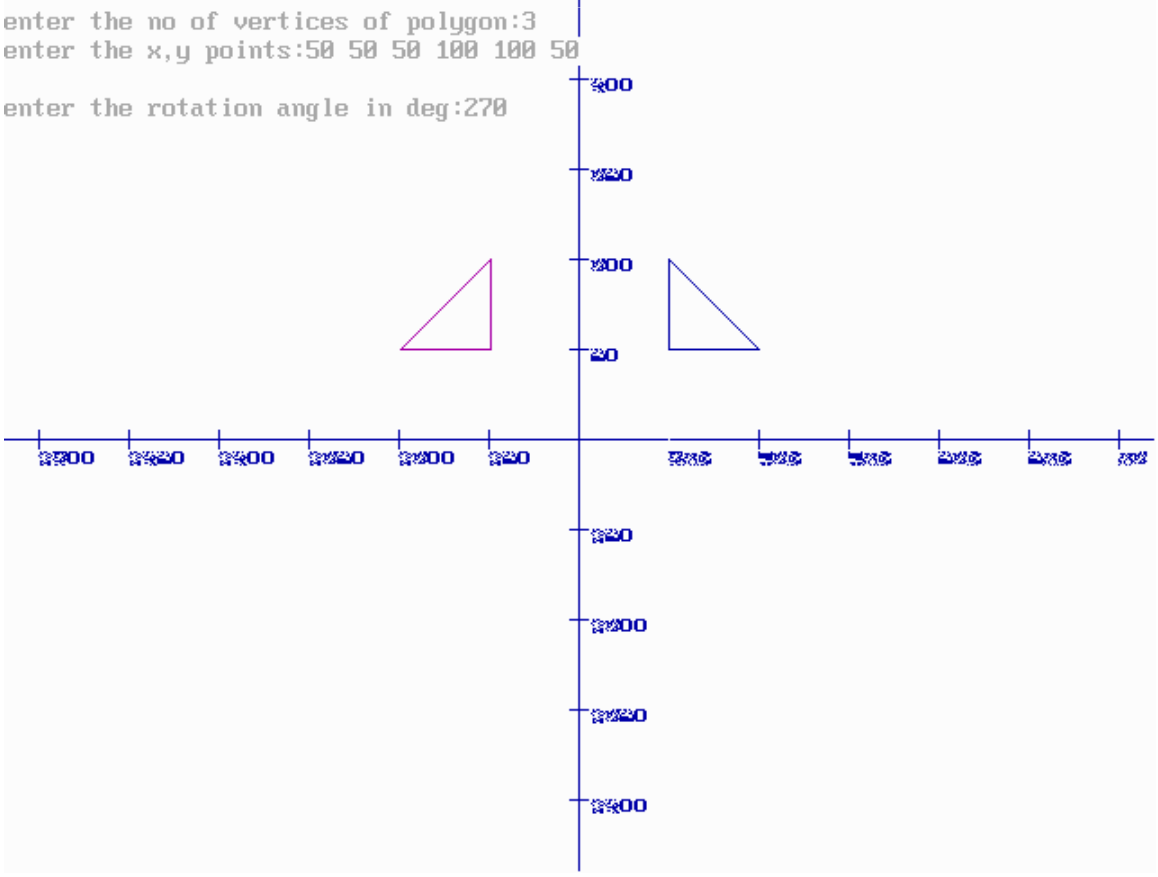
```
                case 4:
                        exit(0);
        }
        for(i=0;i<n-1;i++)
        {
                setcolor(1.5);
                line(x[i]+319,239-y[i],319+x[i+1],239-y[i+1]);
        }
        line(x[i]+319,239-y[i],319+x[0],239-y[0]);
        getch();
        }
    closegraph();
}
void scale()
{
    float sx,sy;
    printf("\nEnter sx,sy values:");
    scanf("%f%f",&sx,&sy);
    for(i=0;i<n;i++)
    {
        x[i]=x[i]*sx;
        y[i]=y[i]*sy;
    }
}
void rotate()
```

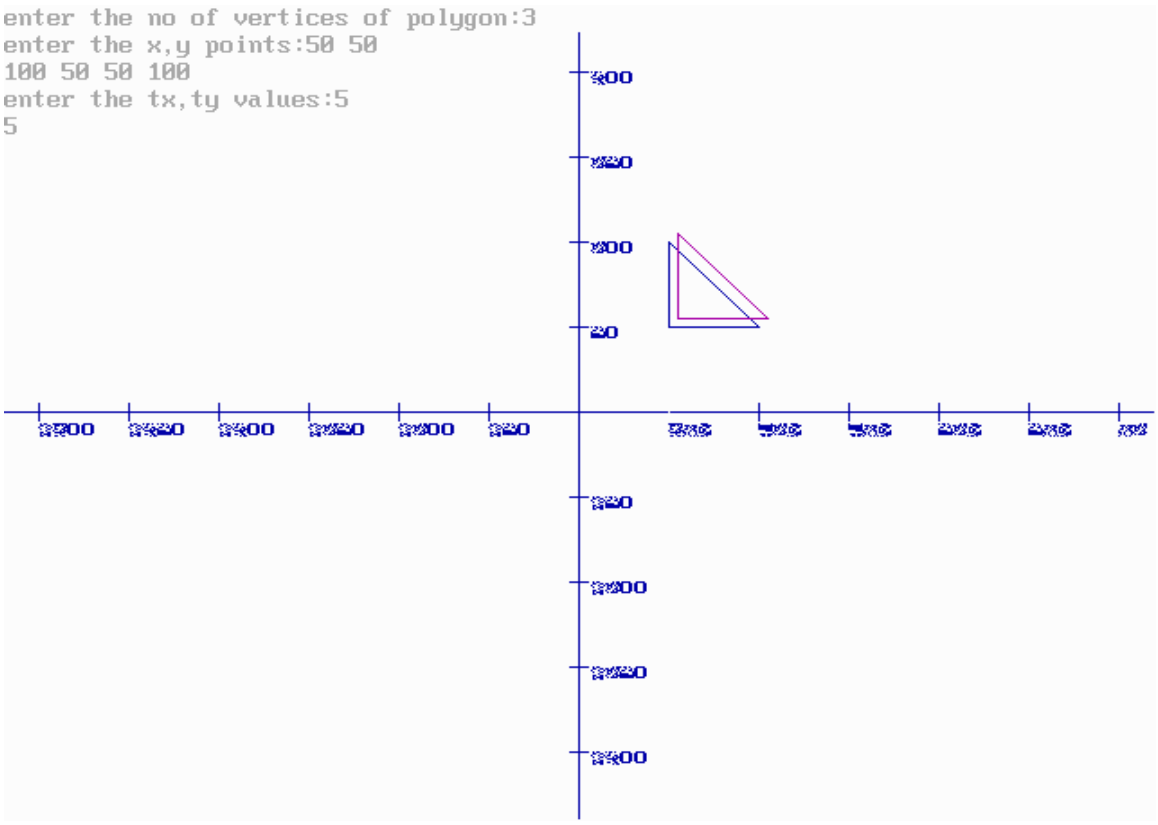
```
{
        float rx,temp;
        printf("\nEnter the rotation angle in degrees:");
        scanf("%f",&rx);
        rx=rx*(22.0/7.0)/180;
        for(i=0;i<n;i++)
        {
                temp=x[i];
                x[i]=(x[i]*cos(rx)+y[i]*sin(rx));
                y[i]=(y[i]*cos(rx)-temp*sin(rx));
        }
}
void transulate()
{
        float tx,ty;
        printf("Enter the tx,ty values:");
        scanf("%f%f",&tx,&ty);
        for(i=0;i<n;i++)
        {
                y[i]=y[i]+ty;
                x[i]=x[i]+tx;
        }
}
```

OUTPUT:

```
 2d-transformations
1:scaling
 2:rotation
 3:translation
 4:exit
enter choice:1
```


enter the no of vertices of polygon:3
enter the x,y points:50 50
100 150 100 200
enter sx,sy values:.5
.5


enter the no of vertices of polygon:3
enter the x,y points:50 50 50 100 100 50
enter the rotation angle in deg:270


enter the no of vertices of polygon:3
enter the x,y points:50 50
100 50 50 100
enter the tx,ty values:5
5

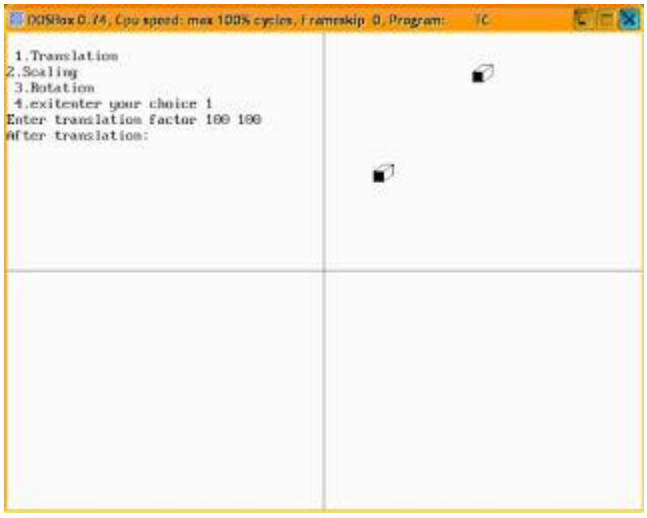
Program 8: Implement 3D transformations like scaling ,rotating and transforming a polygon about an axis.

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
#include<math.h>
int maxx,maxy,midx,midy;
void axis()
{
getch();
cleardevice();
line(midx,0,midx,maxy);
line(0,midy,maxx,midy);
}
void main()
{
int gd,gm,x,y,z,o,x1,x2,y1,y2;
detectgraph(&gd,&gm);
initgraph(&gd,&gm,"d:\\tc\\bgi");
setfillstyle(0,getmaxcolor());
maxx=getmaxx();
maxy=getmaxy();
midx=maxx/2;
midy=maxy/2;
axis();
bar3d(midx+100,midy-150,midx+60,midy-100,10,1);
printf("\Enter the translation factor");
scanf("%d%d",&x,&y);
axis();
printf("After translation");
bar3d(midx+100,midy-150,midx+60,midy-100,10,1);
bar3d(midx+x+100,midy-(y+150),midx+x+60,midy-(y+100),10,1);
axis();
bar3d(midx+100,midy-150,midx+60,midy-100,10,1);
printf("Enter the scaling factor");
scanf("%d%d%d",&x,&y,&z);
axis();
printf("After scaling");
bar3d(midx+100,midy-150,midx+60,midy-100,10,1);
```

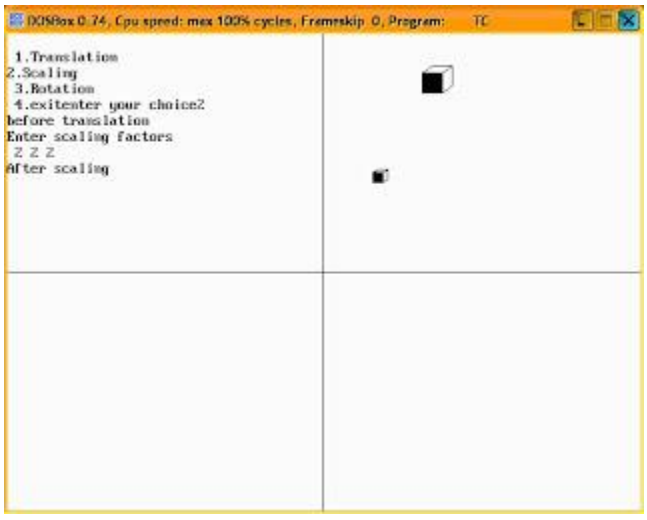
```
bar3d(midx+(x*100),midy-(y*150),midx+(x*60),midy-(y*100),10*z,1);
axis();
bar3d(midx+100,midy-150,midx+60,midy-100,10,1);
printf("Enter the rotation angle");
scanf("%d",&o);
x1=50*cos(o*3.14/180)-100*sin(o*3.14/180);
y1=50*sin(o*3.14/180)+100*cos(o*3.14/180);
x2=60*cos(o*3.14/180)-90*sin(o*3.14/180);
y2=60*sin(o*3.14/180)+90*cos(o*3.14/180);
axis();
printf("After rotating about Z-axis");
bar3d(midx+100,midy-150,midx+60,midy-100,10,1);
bar3d(midx+x1,midy-y1,midx+x2,midy-y2,10,1);
axis();
printf("After rotating about x-axis");
bar3d(midx+100,midy-150,midx+60,midy-100,10,1);
bar3d(midx+100,midy-x1,midx+60,midy-x2,10,1);
axis();
printf("After rotating about Y-axis");
bar3d(midx+100,midy-150,midx+60,midy-100,10,1);
bar3d(midx+x1,midy-150,midx+x2,midy-100,10,1);
getch();
closegraph();
}
```

output :

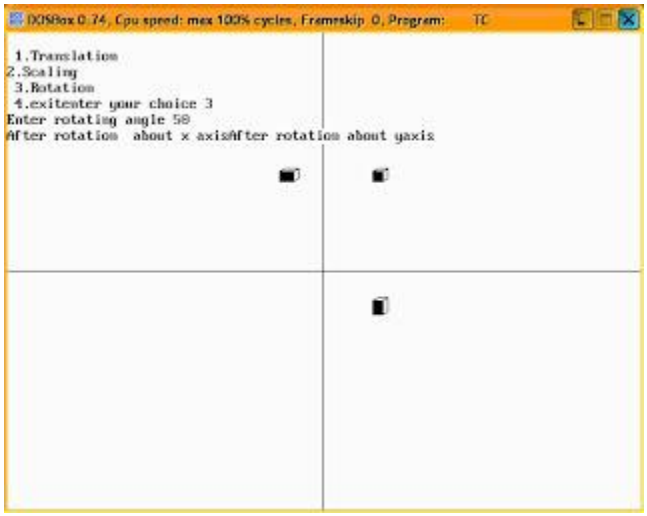
Translation:



Scaling :



Rotation :


DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC
1.Translation
2.Scaling
3.Rotation
4.exitenter your choice 3
Enter rotating angle 50
After rotation about x axisAfter rotation about yaxis

--

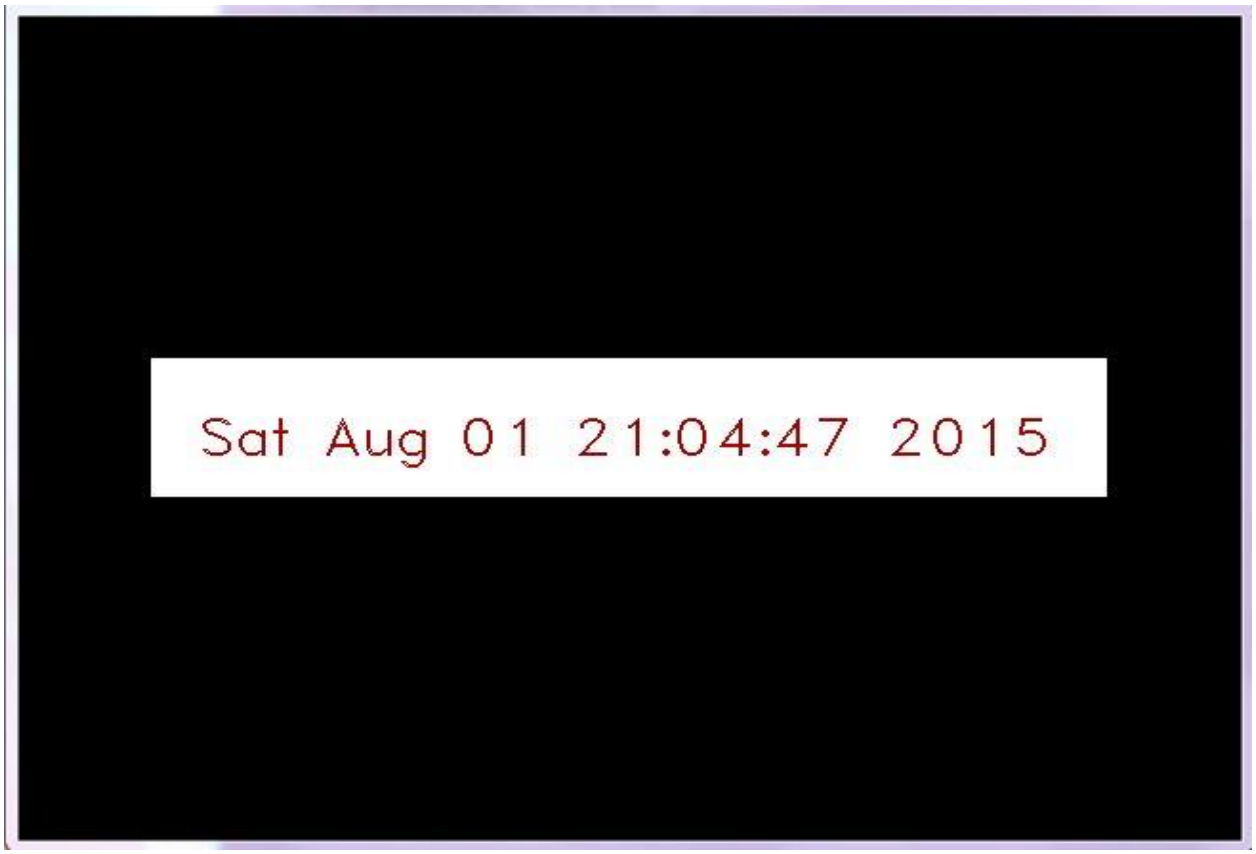
Program 9: Program to make a digital clock

```
#include <conio.h>
#include <graphics.h>
#include <time.h>
#include <dos.h>
#include <string.h>

int main() {
int gd = DETECT, gm;
int midx, midy;
long current_time;
char timeStr[256];

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

/* mid pixel in horizontal and vertical axis */
midx = getmaxx() / 2;
midy = getmaxy() / 2;

while (!kbhit()) {
cleardevice();
setcolor(WHITE);
setfillstyle(SOLID_FILL, WHITE);
rectangle(midx - 250, midy - 40, midx + 250, midy + 40);
floodfill(midx, midy, WHITE);
/* Get Current epoch time in seconds */
current_time = time(NULL);
/* store the date and time in string */
strcpy(timeStr, ctime(&current_time));
setcolor(RED);
settextjustify(CENTER_TEXT, CENTER_TEXT);
settextstyle(SANS_SERIF_FONT, HORIZ_DIR, 4);

moveto(midx, midy);
/* print current time */
outtext(timeStr);
/* Add delay of 1000 milliseconds(1 second) */
delay(1000);
```

```
}

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

Program Output

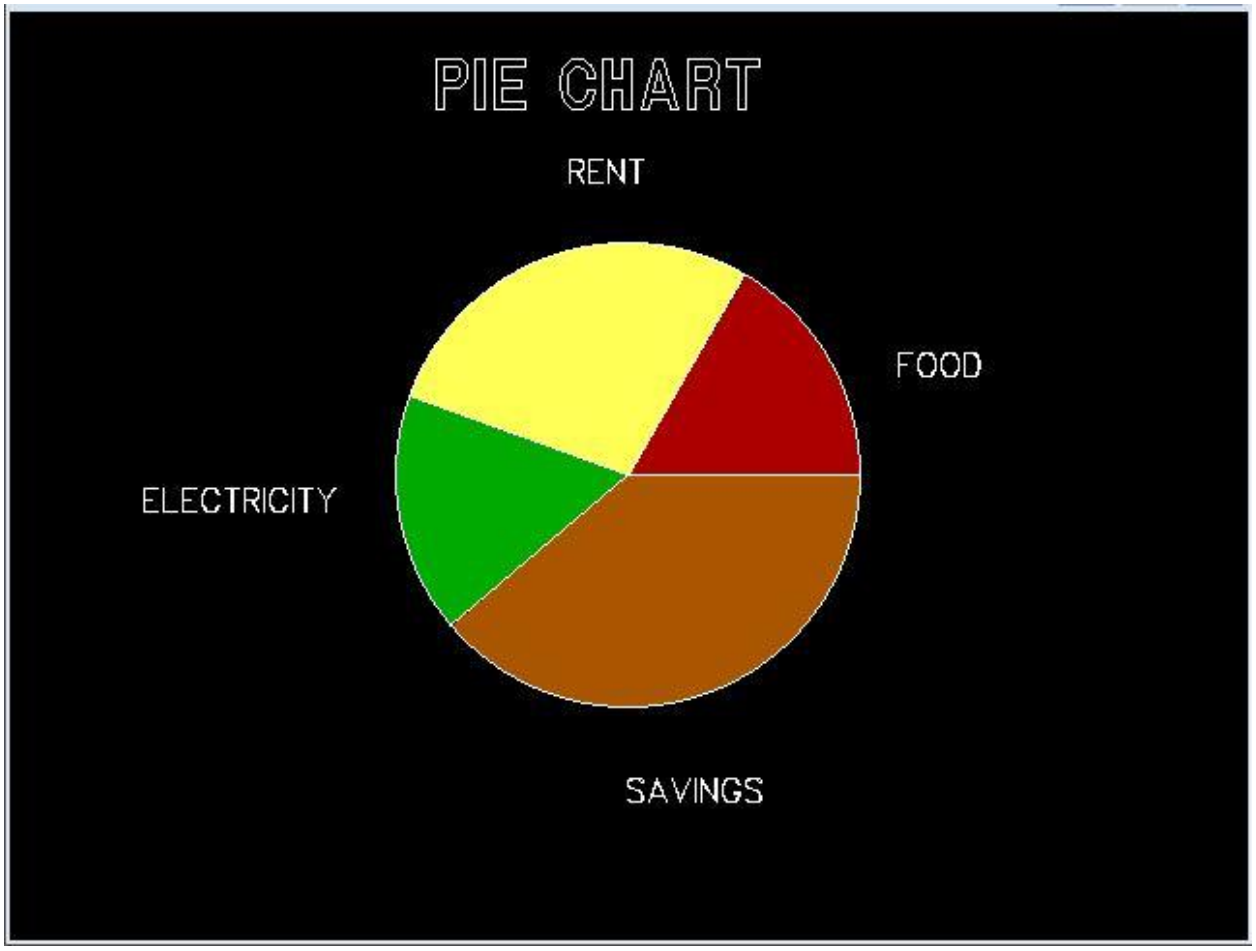
Here is the screen shot of digital clock that shows current date and time on screen.



Program 10. Program to draw a pie chart

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

int main() {
  int gd = DETECT, gm, x, y;
  initgraph(&gd, &gm, "C:\\TC\\BGI");

  settextstyle(BOLD_FONT,HORIZ_DIR,2);
  outtextxy(220,10,"PIE CHART");
  /* Setting cordinate of center of circle */
  x = getmaxx()/2;
  y = getmaxy()/2;

  settextstyle(SANS_SERIF_FONT,HORIZ_DIR,1);
  setfillstyle(SOLID_FILL, RED);
  pieslice(x, y, 0, 60, 120);
  outtextxy(x + 140, y - 70, "FOOD");

  setfillstyle(SOLID_FILL, YELLOW);
  pieslice(x, y, 60, 160, 120);
  outtextxy(x - 30, y - 170, "RENT");

  setfillstyle(SOLID_FILL, GREEN);
  pieslice(x, y, 160, 220, 120);
  outtextxy(x - 250, y, "ELECTRICITY");

  setfillstyle(SOLID_FILL, BROWN);
  pieslice(x, y, 220, 360, 120);
  outtextxy(x, y + 150, "SAVINGS");

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

Program Output



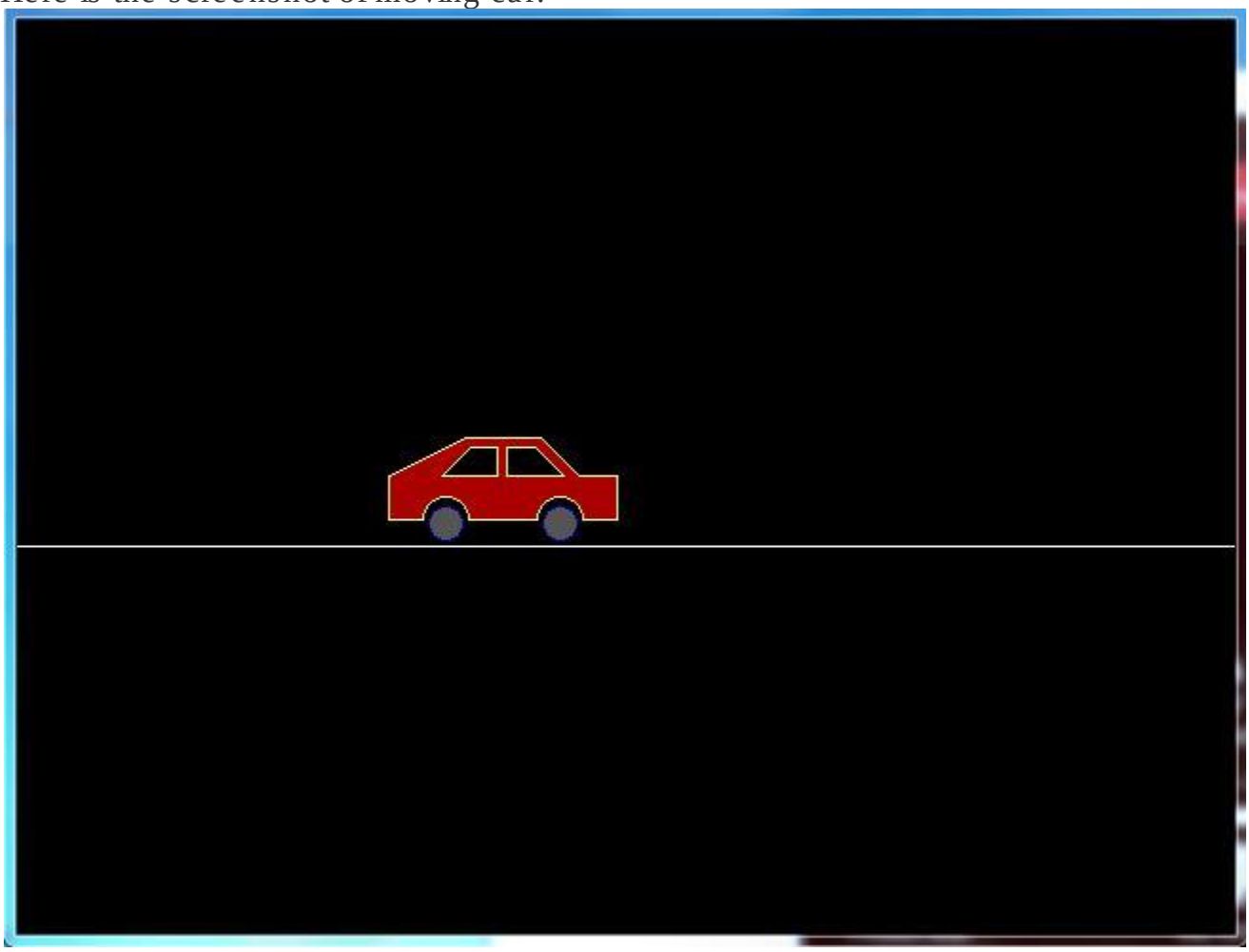
Program 11: Program for moving car animation

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

int main() {
   int gd = DETECT, gm;
   int i, maxx, midy;

   /* initialize graphic mode */
   initgraph(&gd, &gm, "X:\\TC\\BGI");
   /* maximum pixel in horizontal axis */
   maxx = getmaxx();
   /* mid pixel in vertical axis */
   midy = getmaxy()/2;

   for (i=0; i < maxx-150; i=i+5) {
      /* clears screen */
      cleardevice();

      /* draw a white road */
      setcolor(WHITE);
      line(0, midy + 37, maxx, midy + 37);

      /* Draw Car */
      setcolor(YELLOW);
      setfillstyle(SOLID_FILL, RED);

      line(i, midy + 23, i, midy);
      line(i, midy, 40 + i, midy - 20);
      line(40 + i, midy - 20, 80 + i, midy - 20);
      line(80 + i, midy - 20, 100 + i, midy);
      line(100 + i, midy, 120 + i, midy);
      line(120 + i, midy, 120 + i, midy + 23);
      line(0 + i, midy + 23, 18 + i, midy + 23);
      arc(30 + i, midy + 23, 0, 180, 12);
```

```
        line(42 + i, midy + 23, 78 + i, midy + 23);
        arc(90 + i, midy + 23, 0, 180, 12);
        line(102 + i, midy + 23, 120 + i, midy + 23);
        line(28 + i, midy, 43 + i, midy - 15);
        line(43 + i, midy - 15, 57 + i, midy - 15);
        line(57 + i, midy - 15, 57 + i, midy);
        line(57 + i, midy, 28 + i, midy);
        line(62 + i, midy - 15, 77 + i, midy - 15);
        line(77 + i, midy - 15, 92 + i, midy);
        line(92 + i, midy, 62 + i, midy);
        line(62 + i, midy, 62 + i, midy - 15);
        floodfill(5 + i, midy + 22, YELLOW);
        setcolor(BLUE);
        setfillstyle(SOLID_FILL, DARKGRAY);
        /*  Draw Wheels */
        circle(30 + i, midy + 25, 9);
        circle(90 + i, midy + 25, 9);
        floodfill(30 + i, midy + 25, BLUE);
        floodfill(90 + i, midy + 25, BLUE);
        /* Add delay of 0.1 milli seconds */
        delay(100);
    }

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

Program Output

Here is the screenshot of moving car.



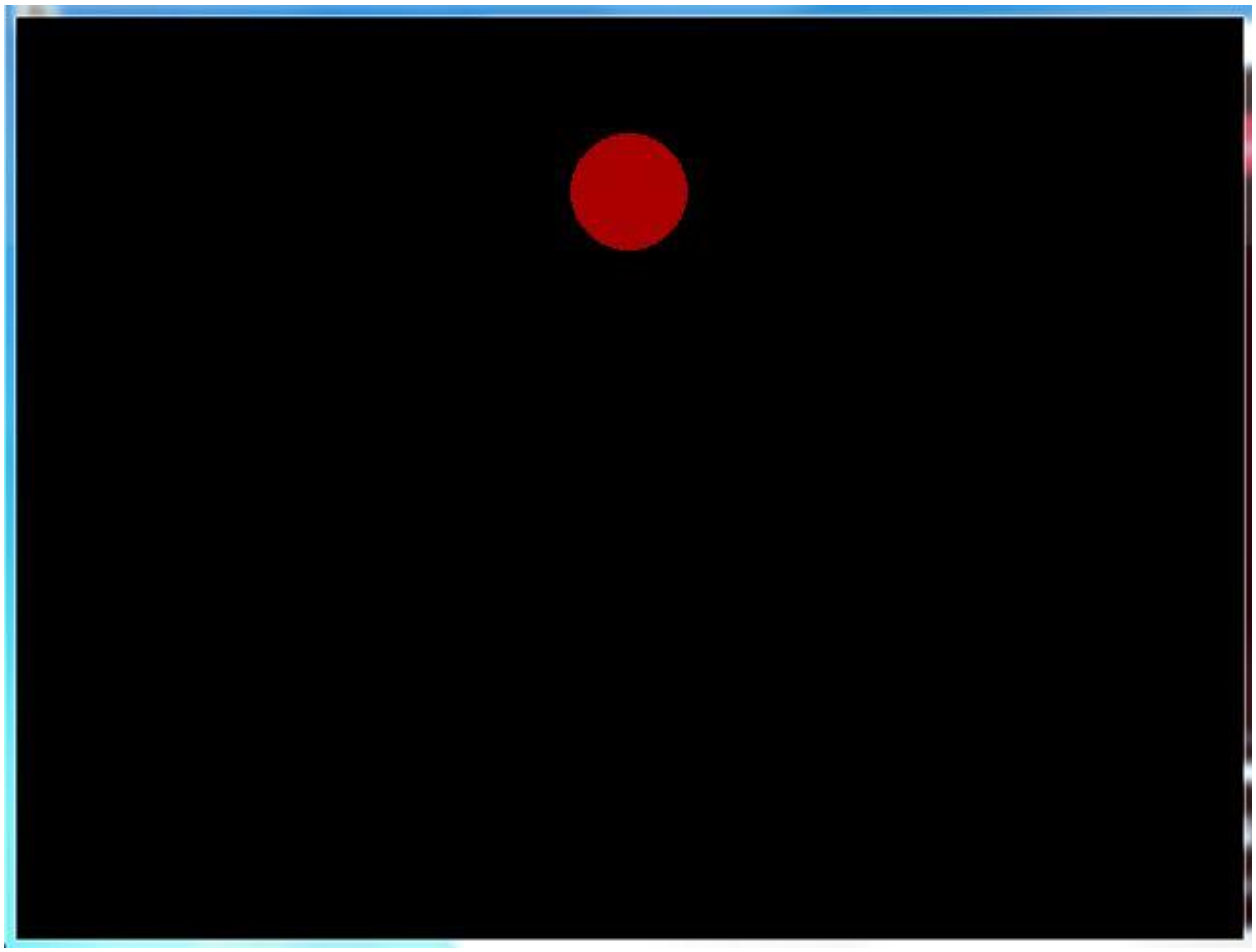
Program 12. Program for bouncing ball animation

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

int main() {
 int gd = DETECT, gm;
 int i, x, y, flag=0;
 initgraph(&gd, &gm, "C:\\TC\\BGI");

 /* get mid positions in x and y-axis */
 x = getmaxx()/2;
 y = 30;


 while (!kbhit()) {
  if(y >= getmaxy()-30 || y <= 30)
     flag = !flag;
     /* draws the gray board */
     setcolor(RED);
     setfillstyle(SOLID_FILL, RED);
     circle(x, y, 30);
     floodfill(x, y, RED);

 /* delay for 50 milli seconds */
 delay(50);

 /* clears screen */
 cleardevice();
 if(flag){
     y = y + 5;
 }
else
{
     y = y - 5;
 }
    }
```

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

Program Output

Here is the screenshot of bouncing ball.



MULTI MEDIA

Program No:1: Create 7-10 slides using a PowerPoint presentation

Steps to create slides:

1.Selecting the text layouts from menu.

2.Selecting the design template form menu.

3.Entering the text on the slides.

Preview animations

On the **Slide Show** menu, click **Custom Animation**, and do one or more of the following:

- To preview animations for a slide, display the slide you want to preview, and then click the **Play** button in the **Custom Animation**

  Note This will show triggered animations automatically; you do not have to click the item to see it animate.

- To preview all animations, including triggered animations, click in the **Custom Animation** task pane.

  Note This will show triggered animations when you click the specified item.

Start a slide show presentation:

- Open the presentation you want to view as a slide show.
- Do one of the following:
  - Click **Slide Show** at the lower left of the PowerPoint window.
  - On the **Slide Show** menu, click **View Show**.
  - Press F5.

OUTPUT:

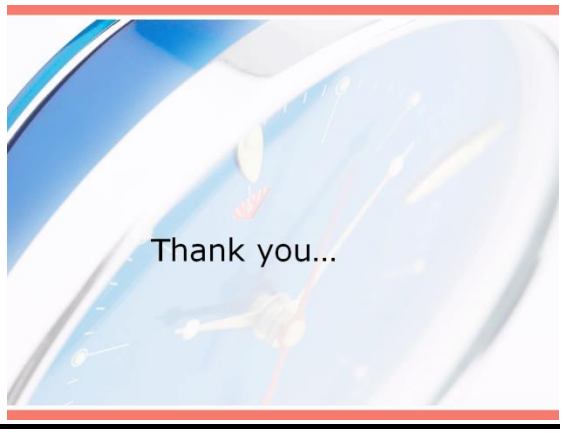

GRAPHICS AND
MULTIMEDIA LAB


Graphics Lab
cycle


List of routines
Creating various types of texts and fonts.
Creating two dimensional objects using the lines and curves.
Animating the two dimensional pictures using transformations.
Coloring the pictures and zooming.
Creating an object and applying Drawing various types of lines and curves.
the animation of heap frame.
Creating 3-dimensional objects using wire frame model.
Rotation, Scaling and Translating the 3-D objects.
Coloring the 3-D objects.
Shading the 3-D objects.
Rendering the objects.
Creating smooth surfaces.
Creating rugged surfaces based on factory geometry.


Multimedia
Lab cycle


List of experiments
Preproduction and presentation of graphics.
Type phases and graphics.
Desktop publishing.
Production planning and design.
User interface design and graphics.
Multimedia sound.
Digital video.
Create the basic web pages using dream viewer/flash.


Thank you...

Program 2: Create Animation using Macro media Flash

Importing a picture:-

Select file -> import

This will place the required picture on the work area.

Creating symbols and instances:-

- Drag the symbol from the library window in to the work area.Then an instance is created.
- Use sale and rotate modifiers of the arrow tool to change instances.

Scaling and Rotatin Picture:-

Scaling:

- Select the picture using arrow tool.
- In the options of arrow tool,select scale options.
- Now drag the scale handles in horizontal/vertical direction,which changes Wthe size of the picture.

Rotation:

- Selecting the arrow tool and select the object .
- Click the rotation modifier in the options.
- Drag one of the corner handles,rotates the picture using the mouse.
- Choose modify>transform>Rotate to hide the rotation handles.

Breaking Apart:

- Select the picture using arrow tool .

- Choose modify>Break a part it changes the gradient of the picture.

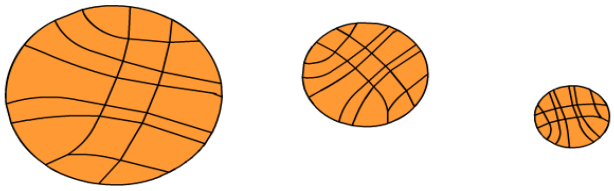
Tweend Animations:-

Tweened animations is a essential technique for creating movement and change in a movie,while,minimizing file size.

Steps to create tweened animations:

- Use the arrow tool to drag a selection boarder around the picture.
- Choose insert > create motion tween.
- Select a frame randomly press F6 to insert a new key frame,drag the picture to the required place on the work area.
- Choose control > play to see the animations.

OUTPUT:



Program 3: Create Audio Files and Practice Editing and Adjusting the Quality.

Steps to create sounds:

1. To record a sound

- 1.Make sure you have an audio input device connected to your computer.
- 2.On the **File** menu, click **New**.
- 3.To begin recording, click **Record** .
- 4.To stop recording, click **Stop** .

- Recorded sounds are saved as waveform (.wav) files.
- You can play your recording in Sound Recorder or in any other program that supports waveform (.wav) files.

2.To adjust the quality of a sound file

- On the **File** menu, click **Open**.
- In the **Open** dialog box, double-click the sound file you want to modify.
- On the **File** menu, click **Properties**.
- Under **Format Conversion**, click the format you want, and then click **Convert Now**.
- Specify the format and attributes you want, and then click **OK**.

- Compressed sound files cannot be edited. Changing the format of a compressed sound file changes the file into an uncompressed file that you can edit.

- You can change the sound quality to untitled, CD, radio, or telephone quality. The CD, radio, and telephone qualities have predefined formats and attributes (for example, sampling frequency and number of channels). However, if you select the **untitled** sound quality, you can specify the format and attributes.

3.To change the format of a sound file

- On the **File** menu, click **Open**.
- In the **Open** dialog box, double-click the sound file you want to modify.
- On the **File** menu, click **Save As**.
- In the **Save As** dialog box, click **Change**.
- In the **Name** list, click the audio format you want.

- You can also add your own format to the **Name** list. To do so, click the new format and attributes you want from the **Format** and **Attributes** lists, and then click **Save As**. Then, in the **Save As** box, type a name for the new format, and click **OK**.
- Sound Recorder uses waveform (.wav) files.

OUTPUT

Format	Size	Size On Disk
ACELP.net1	254KB(260,762 bytes)	256KB(262,144 bytes)
CCITT_ALaw	509KB(521,394 bytes)	512KB(524,288 bytes)
DSP True Speech	509(521,394 bytes)	512KB(524,288 bytes)
GSM 6.10	51.7KB(53,036 bytes)	64.0KB(65,536 bytes)
IMA ADPCM	258KB(264,252 bytes)	272(278,528 bytes)
Microsoft ADPCM	260KB(266,842 bytes)	272(278,528 bytes)
Microsoft G723.1	260KB(266,842 bytes)	272(278,528 bytes)
MPEG Layer 3	223KB(228,550 bytes)	224KB(229,376 bytes)
PCM	5.48MB(5,747,770 bytes)	5.48MB(5,750,784 bytes)
Windows Media Audio-V1	223KB(228,550 bytes)	224 KB(229,376 bytes)
Windows Media Audio-V2	224KB(228,550 bytes)	224KB(229,376 bytes)

Program 4: Create 3 basic web pages using Dream weaver or a HTML package.

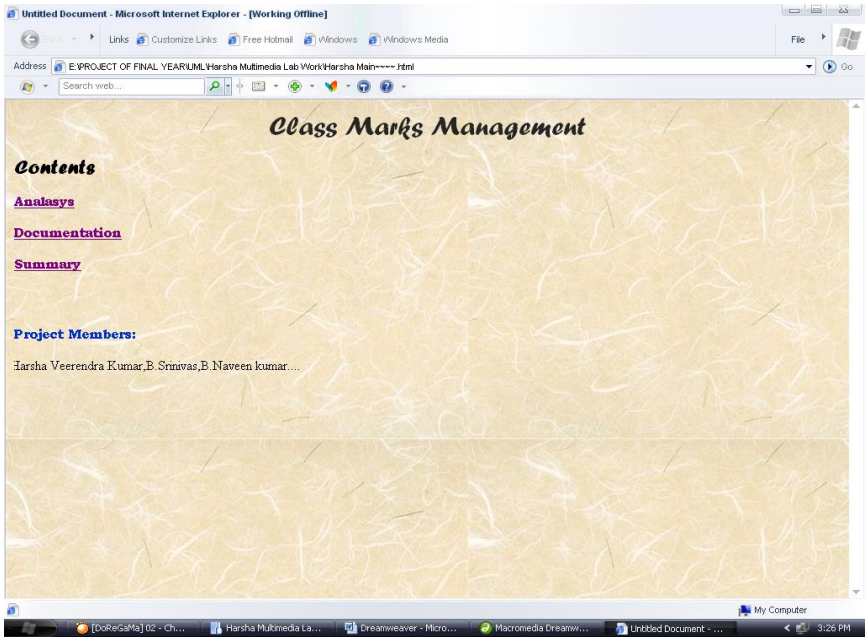
To set HTML formatting options:

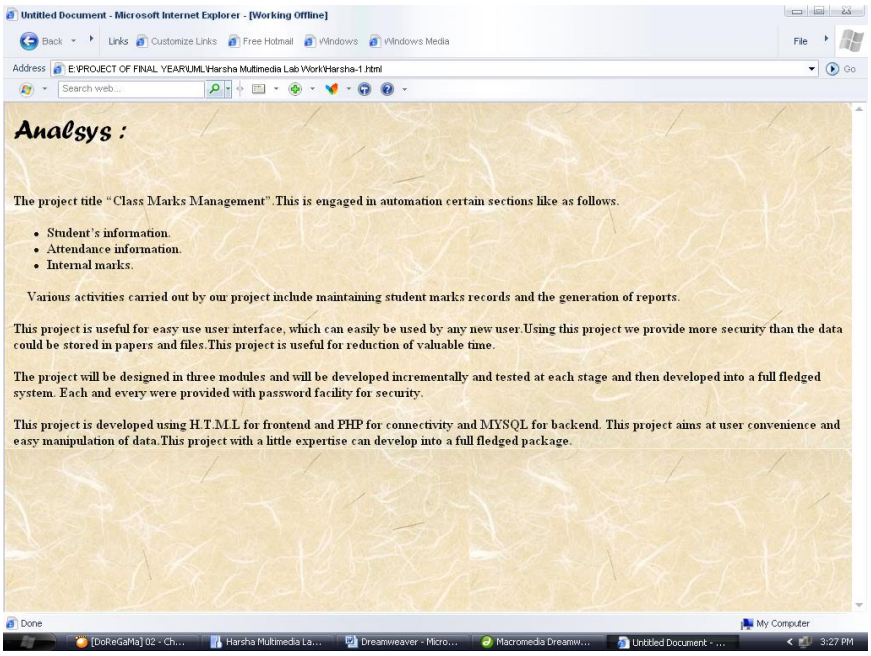
1. Open the Property inspector (Window > Properties), if it isn't already open.
2. Select the text you want to format.
3. Set the options you want to apply to the selected text

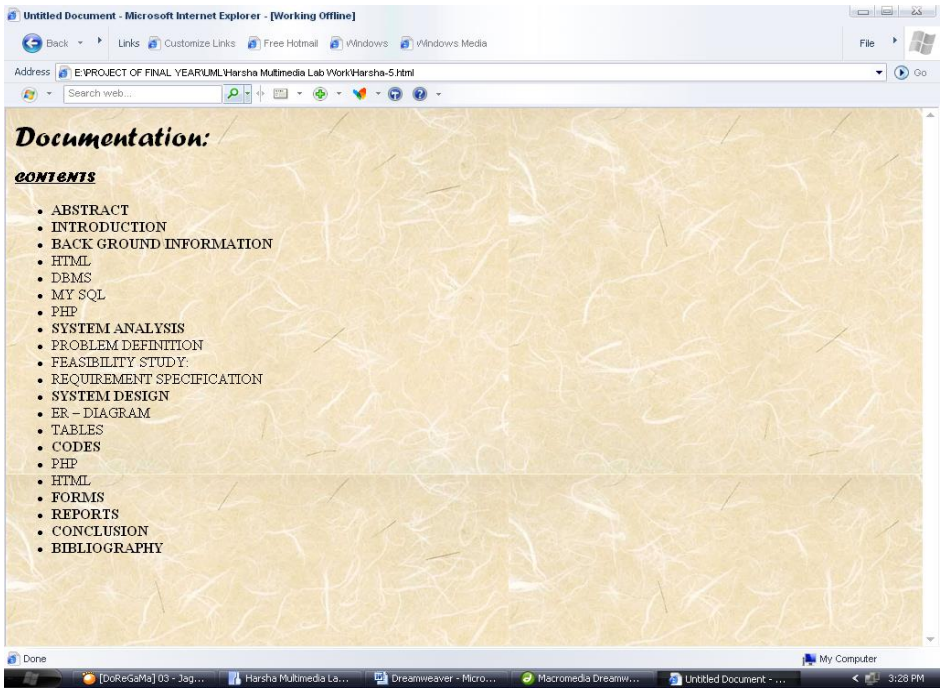
Steps to create an HTML form:

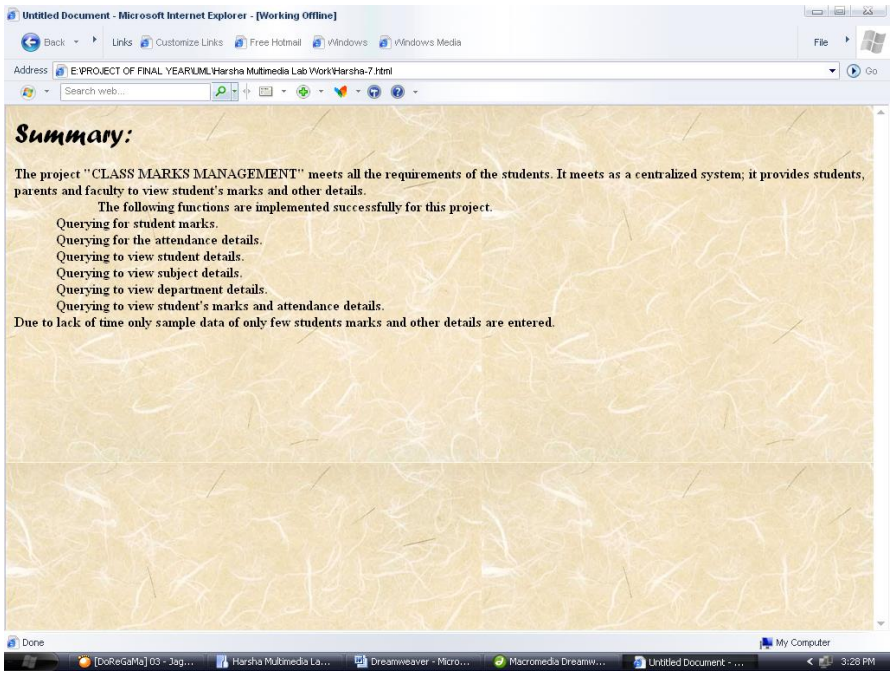
1. Open the Property inspector (Window > Properties), if it isn't already open.
2. Select the text you want to format.
3. Set the options you want to apply to the selected text.

OUTPUT




Untitled Document - Microsoft Internet Explorer - [Working Offline]
Address E:\PROJECT OF FINAL YEAR\UML\Harsha Multimedia Lab Work\Harsha-1.html
Analsys :
The project title "Class Marks Management".This is engaged in automation certain sections like as follows.
Student's information.
Attendance information.
Internal marks.
Various activities carried out by our project include maintaining student marks records and the generation of reports.
This project is useful for easy use user interface, which can easily be used by any new user.Using this project we provide more security than the data could be stored in papers and files.This project is useful for reduction of valuable time.
The project will be designed in three modules and will be developed incrementally and tested at each stage and then developed into a full fledged system. Each and every were provided with password facility for security.
This project is developed using H.T.M.L for frontend and PHP for connectivity and MYSQL for backend. This project aims at user convenience and easy manipulation of data.This project with a little expertise can develop into a full fledged package.
Done
My Computer
3:27 PM


Untitled Document - Microsoft Internet Explorer - [Working Offline]
Address E:\PROJECT OF FINAL YEAR\UML\Harsha Multimedia Lab Work\Harsha-5.html
Documentation:
CONTENTS
ABSTRACT
INTRODUCTION
BACK GROUND INFORMATION
HTML
DBMS
MY SQL
PHP
SYSTEM ANALYSIS
PROBLEM DEFINITION
FEASIBILITY STUDY:
REQUIREMENT SPECIFICATION
SYSTEM DESIGN
ER – DIAGRAM
TABLES
CODES
PHP
HTML
FORMS
REPORTS
CONCLUSION
BIBLIOGRAPHY
Done
My Computer
3:28 PM


Untitled Document - Microsoft Internet Explorer - [Working Offline]
Back
Links
Customize Links
Free Hotmail
Windows
Windows Media
File
Address
E:\PROJECT OF FINAL YEAR\UML\Harsha Multimedia Lab Work\Harsha-7.html
Go
Search web...
Summary:
The project "CLASS MARKS MANAGEMENT" meets all the requirements of the students. It meets as a centralized system; it provides students, parents and faculty to view student's marks and other details.
The following functions are implemented successfully for this project.
Querying for student marks.
Querying for the attendance details.
Querying to view student details.
Querying to view subject details.
Querying to view department details.
Querying to view student's marks and attendance details.
Due to lack of time only sample data of only few students marks and other details are entered.
Done
My Computer
[DoReGaMa] 03 - Jag...
Harsha Multimedia La...
Dreamweaver - Micro...
Macromedia Dreamw...
Untitled Document - ...
3:28 PM

VIVA QUESTIONS

1. Define various types of Graphics?

2. Define Resolution?

3. What is Pixel?

4. Write the Drawbacks of DDA algorithm?

5 . What is the use of Normalized device co ordinates?

6. Differentiate between window and View port?

7. What is Polygon clipping?

8. Define Translation, Scaling and Rotation.

9. What are the various types of image formats?

10. What are the different methods to fill a polygon?

11. What is a transformation?

12. What is clipping?

13. What is a window?

14. What is a viewport?

15. What is the need of homogeneous co ordinate system?

16. What is clip window?

17. Define exterior clipping.

18. Define interior clipping.

19. What is translation?

20. What is scaling?

21. What is rotation?

22. What is reflection?

23. What is shear?

24. What is Primitive?

25. Differentiate between 2D and 3D.

26. Differentiate between raster and random scan display.

27. What is CAD?

28 Define Aspect-ratio.

29. What is meant by frame buffer?

30. What is Grayscale?