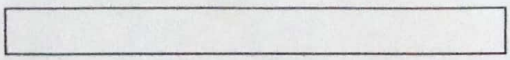
Chapter (2) Image Representation

1.Write a program for Pattern creation using setfillstyle

Solution:

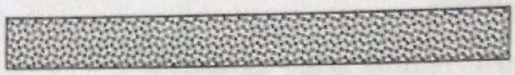
```
#include <graphics.h>
#include <stdlib.h>
#include <string.h>
#include <stdio.h>
#include <conio.h>
void main()
{
int gd=DETECT,gm;
int s;
char *fname[] = { "EMPTY FILL","SOLID FILL","LINE FILL",
"LTSLASH FILL","SLASH FILL","BKSLASH FILL",
"LTBKSLASH FILL","HATCH FILL",
"XHATCH FILL","INTERLEAVE FILL",
"WIDE DOT FILL","CLOSE DOT FILL","USER FILL"
};
initgraph(&gd,&gm,"c:\\turboc3\\BGI ");
clrscr();
cleardevice();
for (s=0;s<13;s++)
{
setfillstyle(s,2);
setcolor(14);
rectangle(149,9+s*30,251,31+s*30);
bar(150,10+s*30,250,30+s*30);
outtextxy(255,20+s*30,fname[s]);
}
getch();
closegraph();
}
```

Output:

Empty fill 

Solid fill 

Close dot fill 

Wide dot fill 

Line fill 

Back Slash fill 

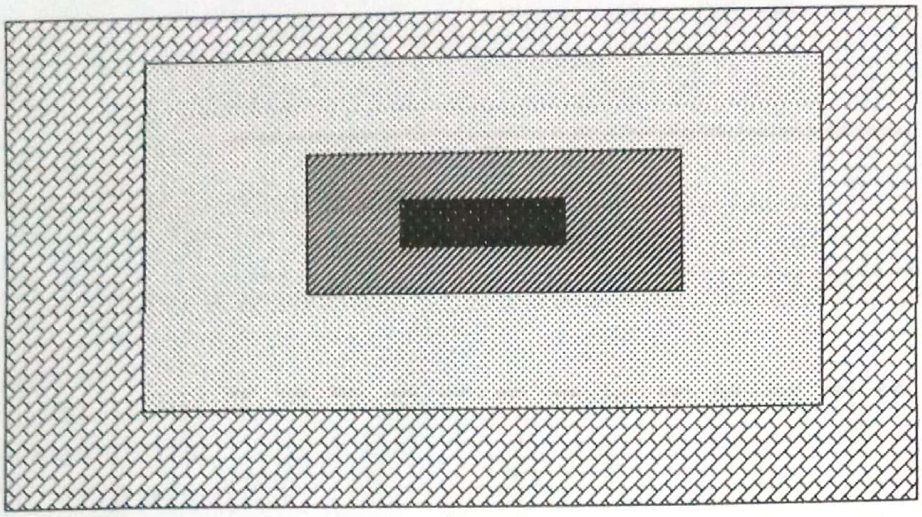
User fill 

2. Write a program for Random Pattern Generation

Solution:

```
#include "graphics.h"
#include "conio.h"
#include "stdlib.h"

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

   initgraph(&gd, &gm, "c:\\tuboc3\\BGI");
   setcolor(3);
   setfillstyle(SOLID_FILL,RED);
   bar(50, 50, 590, 430);

   setfillstyle(1, 14);
   bar(100, 100, 540, 380);

   while(!kbhit())
   {
      putpixel(random(439)+101,  random(279)+101,random(16));
      setcolor(random(16));
      circle(320,240,random(100));
   }
   getch();
   closegraph();
}
```

Output:

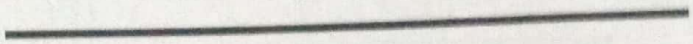


3. Write a program for Line Pattern Generation

Solution:

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

void main()

{
int gd=DETECT,gm;
int s;
char *lname[]={"SOLID LINE","DOTTED LINE","CENTER LINE",
"DASHED LINE","USERBIT LINE"};
initgraph(&gd,&gm,"c:\\turboc3\\BGI ");
clrscr();
cleardevice();
printf("Line styles:");
for (s=0;s<5;s++)
{
setlinestyle(s,1,3);
line(100,30+s*50,250,250+s*50);
outtextxy(255,250+s*50,lname[s]);
}


getch();
closegraph();
}
```

Output:

Solid Line 

Dotted Line 

Dashed Line 

Userbit Line 

Chapter(3): Scan Conversion

1. ***Write a program to draw a line using DDA Line Drawing Algorithm:***

Solution:

```
#include "stdio.h"
        #include "conio.h"
        #include "math.h"
        #include "graphics.h"
        main()
        {
                int gd=DETECT,gm;
                int xa,xb,ya,yb;
                int dx,dy,steps,k,xinc,yinc,x,y;
                initgraph(&gd,&gm,"c:\\turboc3\\bgi");
                printf("Enter the two left end pixel points(xa,ya):\n");
                scanf("%d%d",&xa,&ya);
                printf("Enter the two Right end pixel points(xb,yb):\n");
                scanf("%d%d",&xb,&yb);
                dx=xb-xa;
                dy=yb-ya;
                if(abs(dx)>abs(dy))
                        steps=abs(dx);
                else
                        steps=abs(dy);
                xinc=dx/steps;
                yinc=dy/steps;
                x=xa;
                y=ya;
                putpixel(x,y,6);
                for(k=1;k<=steps;k++)
                {
                        x=x+xinc;
                        y=y+yinc;
                        putpixel(x,y,6);
```

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

Output:

Enter the two left end pixel points(xa,ya): 100, 100

Enter the two Right end pixel points(xb,yb): 200, 200

2. Write a program to draw a dotted line using DDA Line Drawing Algorithm:

Solution:

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
void main()
{
int gd=DETECT,gm,xa,xb,ya,yb,i;
double xin,yin,dx,dy,x,y,steps;
clrscr();
initgraph(&gd,&gm,"C:\\turboc3\\bgi");
printf("Enter Xa=");
scanf("%d",&xa);
printf("Enter Ya=");
scanf("%d",&ya);
printf("Enter Xb=");
scanf("%d",&xb);
printf("Enter Yb=");
scanf("%d",&yb);
cleardevice();
if(xb>xa)
dx=xb-xa;
else
dx=xa-xb;
if(yb>ya)
dy=yb-ya;
else
dy=ya-yb;
if(dx>dy)
steps=dx;
else
steps=dy;
xin=dx/steps;
yin=dy/steps;
x=xa;y=ya;
for(i=0;i<steps;i++)
{
if(i%5==1 || i%5==2 || i%5==3)
putpixel((int)x,(int)y,WHITE);
x=x+xin;
```

```
y=y+yin;
}
getch();
closegraph();
}
```

Output:

Enter Xa=100, Enter Ya=100

Enter Xb=200, Enter Yb=200.

3. Write a program to draw a line using Bresenhams Line Drawing Algorithm:

Solution:

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
void main()
{
int x,y,x1,y1,x2,y2,gx,gy,dx,dy,m,i,p;
clrscr();
printf("Enter the starting point of line\n");
scanf("%d%d",&x1,&y1);
printf("Enter the end point of the line\n");
scanf("%d%d",&x2,&y2);
detectgraph(&gx,&gy);
initgraph(&gx,&gy,"c:\\turboc3\\bgi");
dx=x2-x1;
dy=y2-y1;
p=(2*dy)-dx;
for(i=1;i<dx;i++)
{
if(p<0)
{
x=x1++;
y=y1;
putpixel(x,y,30);
p=p+(2*dy);

}
else
{
x=x1++;
y=y1++;
putpixel(x,y,30);
p=p+(2*dy)-(2*dx);
}
}
getch();
}
```

$x_2' - x_1'$,
$y_2' - y_1'$
$d_1 = 2\Delta y - \Delta x$

→ while $(x < x_2')$

$d < 0$
$x++$
y = no change
$d = d + 2\Delta S$

$x++$
$y++$
$d = d + dT$

Output:
Enter the starting point of line:
Enter the ending point of line:

4. Write a program to draw a circle using Bresenhams Circle Drawing Algorithm:

Solution:

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
int r,x,y;
float a,b,d;
void bresan();
void pixel(float a,float b);
void main()
{
int gm,gr;
clrscr();
detectgraph(&gm,&gr);
initgraph(&gm,&gr,"c:\\turboc3\\BGI");
printf("Enter the center point:\n");
scanf("%d%d",&x,&y);
printf("Enter the radius :\n");
scanf("%d",&r);
bresan();
getch();
}
void bresan()
{
a=0;
b=r;
d=3-(2*r);
while(a<=b)
{
a++;
if(d>=0)
{
b--;
d=d+10+4*(a-b);
}
else
d=d+(4*a)+6;
pixel(a,b);
}
}
void pixel(float a,float b)
{
putpixel(x+a,y+b,10);
```

```
putpixel(x-a,y-b,10);
putpixel(x+a,y-b,10);
putpixel(x-a,y+b,10);
putpixel(x+b,y+a,10);
putpixel(x-b,y-a,10);
putpixel(x+b,y-a,10);
putpixel(x-b,y+a,10);
}
```

Output:

Enter the center point:
Enter the radious:

5. Write a program to draw a circle using MidPoint Circle Drawing Algorithm:

Solution:

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
int x,y,p,r,d,a,b;
void mid();
void pixel(int a,int b);
void main()
{
int gm,gr;
clrscr();
detectgraph(&gm,&gr);
initgraph(&gm,&gr,"c:\\turboc3\\BGI");
printf("Enter the center point:\n");
scanf("%d%d",&a,&b);
printf("Enter the radius:\n");
scanf("%d",&r);
mid();
getch();
}
void mid()
{
x=0;
y=r;
p=1-r;
while(x<y)
{
pixel(a,b);
if(p<0)
x=x+1;
else
{
x=x+1;
```

```
y=y-1;
}
if(p<0)
p=p+2*x+1;
else
p=p+2*(x-y)+1;
}
}
void pixel(int a,int b)
{
putpixel(a+x,b+y,10);
putpixel(a-x,b+y,10);
putpixel(a+x,b-y,10);
putpixel(a-x,b-y,10);
putpixel(a+y,b+x,10);
putpixel(a-y,b-x,10);
putpixel(a+y,b-x,10);
putpixel(a-y,b+x,10);
}
```

Output:

Enter the center point:
Enter the radious:

6. Write a program to draw a ellipse using MidPoint ellipse Drawing Algorithm:

Solution:

```
#include<stdio.h>
#include<graphics.h>
#include<math.h>
void main()
{
long d1,d2;
int i,gd,gm,x,y,x0,y0;
long rx,ry,rxsq,rysq,tworxsq,tworysq,dx,dy;
clrscr();printf("Enter the X radius and Y radius of the ellipse:\n");
scanf("%ld%ld",&rx,&ry);
printf("\nEnter the center (x,y) of the ellipse:\n");
scanf("%d%d",&x0,&y0);
detectgraph(&gd,&gm);
initgraph(&gd,&gm,"c:\\turboc3\\BGI");
cleardevice();
rxsq=rx*rx;
rysq=ry*ry;
tworxsq=2*rxsq;
tworysq=2*rysq;
```

```
x=0;
y=ry;
d1=rysq-rxsq*ry+(0.25*rxsq);
dx=tworysq*x;
dy=tworxsq*y;
do
{
putpixel(x0+x,y0+y,15);
putpixel(x0-x,y0-y,15);
putpixel(x0+x,y0-y,15);
putpixel(x0-x,y0+y,15);
if(d1<0)
{
x=x+1;
y=y;
dx=dx+tworysq;
d1=d1+dx+rysq;
}
else
{
x=x+1;
y=y-1;
dx=dx+tworysq;
dy=dy-tworxsq;
d1=d1+dx-dy+rysq;
}
delay(10);
}
while(dx<dy);
d2=rysq*(x+0.5)*(x+0.5)+rxsq*(y-1)*(y-1)-(rxsq*rysq);
do
{
putpixel(x0+x,y0+y,15);
putpixel(x0-x,y0-y,15);
putpixel(x0+x,y0-y,15);
putpixel(x0-x,y0+y,15);
if(d2>0)
{
x=x;
y=y-1;
dy=dy-tworxsq;
d2=d2-dy+rxsq;
}
else
{
x=x+1;
y=y-1;
dx=dx+tworysq;
```

```
dy=dy-tworxsq;
d2=d2+dx-dy+rxsq;
}
}
while(y>0);
getch();
closegraph();
}
```

Output:

Enter the X radius and Y radius of the ellipse:

Enter the center (x,y) of the ellipse:

Chapter-4(Two Dimentional Transformation)

1. Write a program to implement 2D Translation of Rectangle:

Solution:

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

void RectAngle(int x,int y,int Height,int Width);
void Translate(int x,int y,int Height,int Width);

void main()
{
   int gd=DETECT,gm;
   int x,y,Height,Width;
   initgraph(&gd,&gm,"c:\\turboc3\\bgi ");
   printf("Enter the First point for the Rectangle:");
   scanf("%d%d",&x,&y);
   printf("Enter the Height&Width for the Rectangle:");
   scanf("%d%d",&Height,&Width);
   RectAngle(x,y,Height,Width);
  getch();
  cleardevice();
  Translate(x,y,Height,Width);
  RectAngle(x,y,Height,Width);
  getch();
}

void RectAngle(int x,int y,int Height,int Width)
{
   line(x,y,x+Width,y);
   line(x,y,x,y+Height);
   line(x+Width,y,x+Width,y+Height);
   line(x,y+Height,x+Width,y+Height);
```

```
}
void Translate(int x,int y,int Height,int Width)
{
   int Newx,Newy,a,b;
   printf("Enter the Transaction coordinates");
   scanf("%d%d",&Newx,&Newy);
   cleardevice();
   a=x+Newx;
   b=y+Newy;
   RectAngle(a,b,Height,Width);
}
```

Output:

Enter the First point for the Rectangle:
Enter the Height& Width for the Rectangle:

2. **Write a program to implement 2D traformation of Rectangle(1.Translation 2.Rotation 3..Scaling 4.Reflection 5.Shearing):**

Solution:

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
#include<graphics.h>
int ch,x,y,az,i,w,ch1,ch2,xa,ya,ra,a[10],b[10],da,db;
float x1,y1,az1,w1,dx,dy,theta,x1s,y1s,sx,sy,a1[10],b1[10];
void main()
{
int gm ,gr;
clrscr();
detectgraph(&gm,&gr);
initgraph(&gm,&gr,"c:\\turboc3\\BGI");
printf("Enter the upper left corner of the rectangle:\n");
scanf("%d%d",&x,&y);
printf("Enter the lower right corner of the rectangle:\n");
scanf("%d%d",&az,&w);
rectangle(x,y,az,w);
da=az-x;
db=w-y;
a[0]=x;
b[0]=y;
a[1]=x+da;
b[1]=y;
a[2]=x+da;
b[2]=y+db;
a[3]=x;b[3]=y+db;
while(1)
{
printf("******2D Transformations*******\n");
printf("1.Translation\n2.Rotation\n3.Scaling\n4.Reflection\n5.Shearing\n6.Exit\nEnter your
choice:\n");
scanf("%d",&ch);
switch(ch)
{
```

```
case 1:
detectgraph(&gm,&gr);
initgraph(&gm,&gr,"d:\\tc\\BGI");
rectangle(x,y,az,w);
printf("*******Translation*******\n\n");
printf("Enter the value of shift vector:\n");
scanf("%f%f",&dx,&dy);
x1=x+dx;
y1=y+dy;
az1=az+dx;
w1=w+dy;
rectangle(x1,y1,az1,w1);
break;
case 2:
detectgraph(&gm,&gr);
initgraph(&gm,&gr,"d:\\tc\\BGI");
rectangle(x,y,az,w);
printf("*******Rotation*******\n\n");
printf("Enter the value of fixed point and angle of rotation:Enter the value of fixed point and angle of
rotation:\n");
scanf("%d%d%d",&xa,&ya,&ra);
theta=(float)(ra*(3.14/180));
for(i=0;i<4;i++)
{
a1[i]=(xa+((a[i]-xa)*cos(theta)-(b[i]-ya)*sin(theta)));
b1[i]=(ya+((a[i]-xa)*sin(theta)+(b[i]-ya)*cos(theta)));
}
for(i=0;i<4;i++)
{
if(i!=3)
line(a1[i],b1[i],a1[i+1],b1[i+1]);
else
line(a1[i],b1[i],a1[0],b1[0]);
}
break;
case 3:
detectgraph(&gm,&gr);
initgraph(&gm,&gr,"d:\\tc\\BGI");
rectangle(x,y,az,w);
printf("********Scaling*******\n\n");
printf("Enter the value of scaling factor:\n");
scanf("%f%f",&sx,&sy);
x1=x*sx;
y1=y*sy;
az1=az*sx;
w1=w*sy;
rectangle(x1,y1,az1,w1);
break;
case 4:
detectgraph(&gm,&gr);
initgraph(&gm,&gr,"d:\\tc\\BGI");
rectangle(x,y,az,w);
printf("*******Reflection*********\n");
printf("1.About x-axis\n2.About y-axis\n3.About both axis\nEnter your choice:\n");
scanf("%d",&ch1);
```

```
switch(ch1)
{
case 1:
printf("Enter the fixed point\n");
scanf("%d%d",&xa,&ya);
theta=(float)(90*(3.14/180));
for(i=0;i<4;i++)
{
a1[i]=(xa+((a[i]-xa)*cos(theta)-(-b[i]-ya)*sin(theta)));
b1[i]=(ya+((a[i]-xa)*sin(theta)+(-b[i]-ya)*cos(theta)));
}
for(i=0;i<4;i++)
{
if(i!=3)
line(a1[i],b1[i],a1[i+1],b1[i+1]);
else
line(a1[i],b1[i],a1[0],b1[0]);
}
break;
case 2:
printf("Enter the fixed point\n");
scanf("%d%d",&xa,&ya);
theta=(float)(270*(3.14/180));
for(i=0;i<4;i++)
{
a1[i]=(xa+((-a[i]-xa)*cos(theta)-(b[i]-ya)*sin(theta)));
b1[i]=(ya+((-a[i]-xa)*sin(theta)+(b[i]-ya)*cos(theta)));
}
for(i=0;i<4;i++)
{
if(i!=3)
line(a1[i],b1[i],a1[i+1],b1[i+1]);
else
line(a1[i],b1[i],a1[0],b1[0]);
}
break;
case 3:
printf("Enter the fixed point\n");
scanf("%d%d",&xa,&ya);
theta=(float)(180*(3.14/180));
for(i=0;i<4;i++)
{
a1[i]=(xa+((-a[i]-xa)*cos(theta)-(-b[i]-ya)*sin(theta)));
b1[i]=(ya+((-a[i]-xa)*sin(theta)+(-b[i]-ya)*cos(theta)));
}
for(i=0;i<4;i++)
{
if(i!=3)
line(a1[i],b1[i],a1[i+1],b1[i+1]);
else
line(a1[i],b1[i],a1[0],b1[0]);
}
break;
}
break;
```

```
case 5:
detectgraph(&gm,&gr);
initgraph(&gm,&gr,"c:\\turboc3\\BGI");
rectangle(x,y,az,w);
printf("*******Shearing******\n\n");
printf("1.x-direction shear\n2.y-direction shear\nEnter your choice:\n");
scanf("%d",&ch2);
switch(ch2)
{
case 1:
printf("Enter the value of shear:\n");
scanf("%f",&x1s);
x1=x+(y*x1s);
y1=y;
az1=az+(w*x1s);
w1=w;
rectangle(x1,y1,az1,w1);
break;
case 2:
printf("Enter the value of shear:\n");
scanf("%f",&y1s);
x1=x;
y1=y+(x*y1s);
az1=az;
w1=w+(az*y1s);
rectangle(x1,y1,az1,w1);
break;
}
break;
case 6:
exit(0);
}
getch();
}
}
```

Output:

```
Enter the upper left corner of the rectangle:
100
100
Enter the lower right corner of the rectangle:
200
200
******2DTransformations*******
1.Translation
2.Rotation
3.Scaling
4.Reflection
5.Shearing
6.Exit

Enter your Choice:
```

3. **Write a program to implement 2D traformation of Triangle(1.Translation 2.Rotation 3..Scaling 4.Reflection 5.Shearing):**

Solution:

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
#include<conio.h>
#include<graphics.h>
float tx,ty,sx,sy,r;
float x[15][15],y[15][15];
int i,j=2,choice;
void transform();
void rotate();
void scale();
void transform()
{
        x[i][j-1]=x[i][j-1]+tx;
        x[i][j]=x[i][j]+tx;
        y[i][j-1]=y[i][j-1]+ty;
        y[i][j]=y[i][j]+ty;
}

void rotate()
{
        x[i][j-1]=(x[i][j-1]*cos(r))-(y[i][j-1]*sin(r));
        y[i][j-1]=(x[i][j-1]*sin(r))+(y[i][j-1]*cos(r));
        x[i][j]=(x[i][j]*cos(r))-(y[i][j]*sin(r));
        y[i][j]=(x[i][j]*sin(r))+(y[i][j]*cos(r));
}
void scale()
{
        x[i][j-1]=x[i][j-1]*sx;
        x[i][j]=x[i][j]*sx;
        y[i][j-1]=y[i][j-1]*sy;
        y[i][j]=y[i][j]*sy;
}
int main()
{
        int gd=DETECT,gm;
        initgraph(&gd,&gm,"c:\\turboc3\\bgi");
        x[1][1]=100,x[1][2]=200,y[1][1]=200,y[1][2]=300;
        x[2][1]=200,x[2][2]=200,y[2][1]=300,y[2][2]=200;
        x[3][1]=100,x[3][2]=200,y[3][1]=200,y[3][2]=200;
        textattr(0);
        for(i=1;i<=3;i++)
                line(x[i][j-1],y[i][j-1],x[i][j],y[i][j]);
        while(choice!=4)
        {
          printf("2D COMPOSITE");
          printf("\n1.Translation & Rotation");
          printf("\n2. Translation & Scaling");
          printf("\n3.Scaling & Rotation");
          printf("\n4.exit");
```

```
        printf("\nEnter your choice:");
        scanf("%d",&choice);
        switch(choice)
        {
                case 1:
                        printf("\nEnter X & Y vector:");
                        scanf("%f%f",&tx,&ty);
                        printf("\nEnter Rotation Angle:");
                        scanf("%f",&r);
                        clrscr();
                        r=(r*3.14/180);
                        for(i=1;i<=3;i++)
                        {
                                transform();
                                rotate();
                                line(x[i][j-1],y[i][j-1],x[i][j],y[i][j]);
                                delay(500);
                        }
                        break;
                case 2:
                        printf("\nEnter X & Y vector:");
                        scanf("%f%f",&tx,&ty);
                        printf("\nEnter X & Y vector:");
                        scanf("%f%f",&sx,&sy);
                        clrscr();
                        for(i=1;i<=3;i++)
                        {
                                transform();
                                scale();
                                line(x[i][j-1],y[i][j-1],x[i][j],y[i][j]);
                                delay(500);
                        }
                        break;
                case 3:
                        printf("\nEnter X & Y vector:");
                        scanf("%f%f",&sx,&sy);
                        printf("\nEnter Rotation Angle:");
                        scanf("%f",&r);
                        clrscr();
                        r=(r*3.14/180);
                        for(i=1;i<=3;i++)
                        {
                                scale();
                                rotate();
                                line(x[i][j-1],y[i][j-1],x[i][j],y[i][j]);
                                delay(500);
                        }
                        break;
                case 4:
                        exit(0);
        }
}
        getch();
        return 0;
}
```

Solution:

*****2D COMPOSITE***

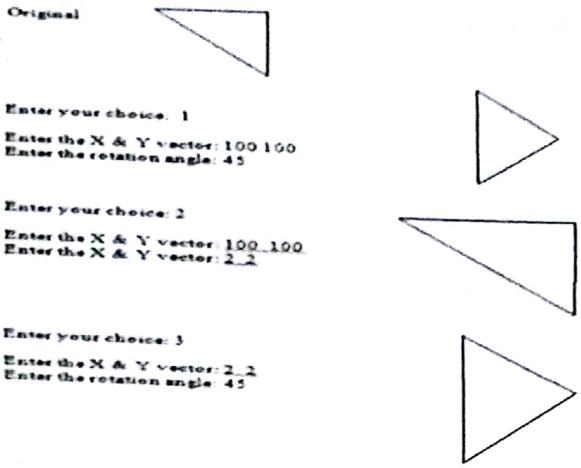
1.Translation & Rotation:

2. Translation & Scaling:

3.Scaling & Rotation:

4.exit:

Enter your choice:



Chapter-5 (Two Dimentional Viewing and Clipping):

1. Write a program to implement Windows to viewport mapping:

Solution:

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
#include<math.h>
main()
{
float sx,sy;
int w1,w2,w3,w4,x1,x2,x3,x4,y1,y2,y3,y4,v1,v2,v3,v4;
int gd=DETECT,gm;
initgraph(&gd,&gm,"C:/turboc3/bgi");
printf("Enter the Co-ordinates x1,y1,x2,y2,x3,y3\n");
scanf("%d%d%d%d%d%d",&x1,&y1,&x2,&y2,&x3,&y3);
cleardevice();
w1=5;
w2=5;
w3=635;
w4=465;
rectangle(w1,w2,w3,w4);
line(x1,y1,x2,y2);
line(x2,y2,x3,y3);
getch();
v1=425;
v2=75;
v3=550;
```

```
v4=250;
sx=(float)(v3-v1)/(w3-w1);
sy=(float)(v4-v2)/(w4-w2);
rectangle(v1,v2,v3,v4);
x1=v1+floor(((float)(x1-w1)*sx)+0.5);
x2=v1+floor(((float)(x2-w1)*sx)+0.5);
x3=v1+floor(((float)(x3-w1)*sx)+0.5);
y1=v2+floor(((float)(y1-w2)*sy)+0.5);
y2=v2+floor(((float)(y2-w2)*sy)+0.5);
y3=v2+floor(((float)(y3-w2)*sy)+0.5);
line(x1,y1,x2,y2);
line(x2,y2,x3,y3);
line(x3,y3,x1,y1);
getch();
return 0;
}
```

Output:

Enter the Co-ordinates x1,y1,x2,y2,x3,y3:

2. Cohen Sutherland Line Clipping Algorithm in C Program

Solution:

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
#include<math.h>
void clip(float,float,float);
int i,j=0,n;
int rx1,rx2,ry1,ry2;
float x1[8],y1[8];
void main()
{
int gd=DETECT,gm;
int i,n;
float x[8],y[8],m;
clrscr();
initgraph(&gd,&gm,"");
printf("coordinates for rectangle : ");
scanf("%d%d%d%d",&rx1,&ry1,&rx2,&ry2);
printf("no. of sides for polygon : ");
scanf("%d",&n);
printf("coordinates : ");
for(i=0;i<n;i++)
{
scanf("%f%f",&x[i],&y[i]);
}
```

```
cleardevice();
outtextxy(10,10,"Before clipping");
outtextxy(10,470,"Press any key....");
rectangle(rx1,ry1,rx2,ry2);
for(i=0;i<n-1;i++)
line(x[i],y[i],x[i+1],y[i+1]);
line(x[i],y[i],x[0],y[0]);
getch();
cleardevice();
for(i=0;i<n-1;i++)
{
m=(y[i+1]-y[i])/(x[i+1]-x[i]);
clip(x[i],y[i],m);
}
clip(x[0],y[0],m);
outtextxy(10,10,"After clipping");
outtextxy(10,470,"Press any key....");
rectangle(rx1,ry1,rx2,ry2);
for(i=0;i<j-1;i++)
line(x1[i],y1[i],x1[i+1],y1[i+1]);
getch();
}

void clip(float e,float f,float m)
{
while(e<rx1 && e>rx2 && f<ry1 && f>ry2)
{
if(e<rx1)
{
f+=m*(rx1-e);
e=rx1;
}
else if(e>rx2)
{
f+=m*(rx2-e);
e=rx1;
}
if(f<ry1)
{
e+=(ry1-f)/m;
f=ry1;
}
else if(f>ry2)
{
e+=(ry2-f)/m;
f=ry2;
}
x1[j]=e;
y1[j]=f;
j++;
}
}
```

Output:
coordinates for rectangle :
no. of sides for polygon :

3. Write a program to Implement Point clipping

Solution:

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
void main()
{
int gm,gr,xcmin,ycmin,xcmax,ycmax,x,y,c;
clrscr();
detectgraph(&gm,&gr);
initgraph(&gm,&gr,"c:\\turboc3\\BGI");
printf("Enter the clipmin coordinate :\n");
scanf("%d%d",&xcmin,&ycmin);
printf("Enter the clipmax coordinate :\n");
scanf("%d%d",&xcmax,&ycmax);
rectangle(xcmin,ycmax,xcmax,ycmin);
printf("Enter the coordinate of the point:\n");
scanf("%d%d",&x,&y);
detectgraph(&gm,&gr);
initgraph(&gm,&gr,"c:\\turboc3\\BGI");
putpixel(x,y,15);
printf("\n1.Point clipping\n2.Exit\nEnter your choice:\n");
scanf("%d",&c);
switch(c)
{
case 1:
detectgraph(&gm,&gr);
initgraph(&gm,&gr,"c:\\turbocc\\BGI");
rectangle(xcmin,ycmax,xcmax,ycmin);
printf("*******POINT CLIPPING******\n");
if((xcmin<x) && (x<xcmax))
{
if((ycmin<y) && (y<ycmax))
{
printf("The point is inside the clip window\n");
putpixel(x,y,15);
}
}
else
printf("The point is outside the clipwindow \nThe point is clipped\n");
break;
case 2:
```

```
exit(0);
}
getch();
}
```

OUTPUT

```
Enter the clipmin coordinate :
200
200
Enter the clipmax coordinate :
300
300
Enter the coordinate of the point:
250
250
1.Point clipping
2.Exit
Enter your choice:
1
*******POINT CLIPPING******
The point is inside the clip window
```

Chapter(6): Three Dimensional Transformation

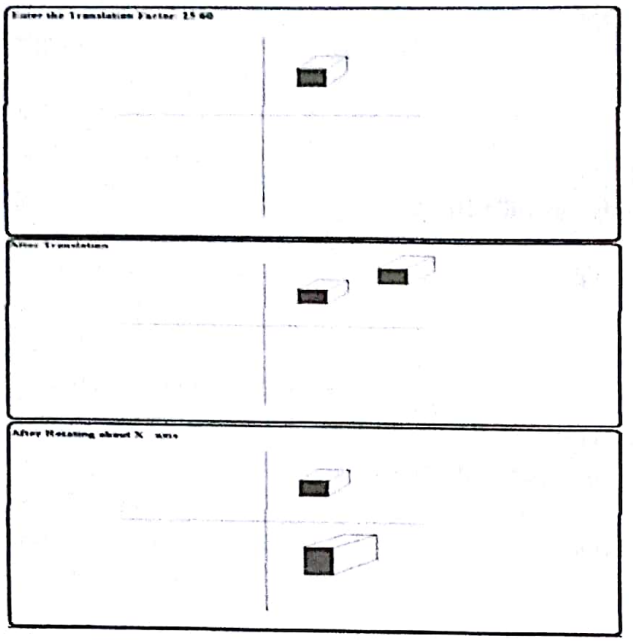
1. Write a program to implement Three Dimensional Transformation Algorithms :

Solution:

```
#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,"c:\\turboc3\\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:

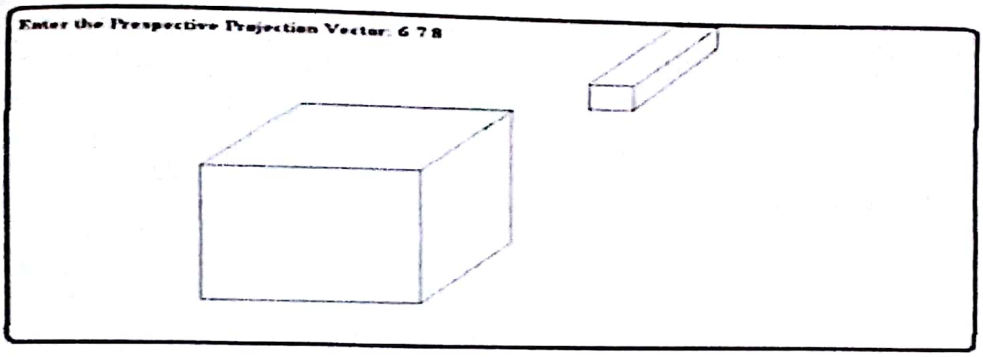


2. Write a *program to implement Visualization of 3-Dimensional Algorithm*

Solution:

```
#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;
clrscr();
detectgraph(&gd,&gm);
initgraph(&gd,&gm,"c:/turboc3/bgi");
setfillstyle(0,getmaxcolor());
maxx=getmaxx();
maxy=getmaxy();
midx=maxx/2;
midy=maxy/2;
bar3d(midx-250,midy+150,midx-150,midy+225,20,4);
printf("\nEnter the Perspective Projection Vector");
scanf("%d%d%d",&x,&y,&z);
bar3d(midx-(x*5),midy-(y*9),midx-(x*3),midy-(y*5),10*z,4);
getch();
closegraph();
}
```

Output:



Others:

1. **Write a program to animate text:**

Solution:

```
#include<stdio.h>
#include<math.h>
#include<conio.h>
#include<graphics.h>
#define round(val) (int)(val+0.5)
void main()
{   int gd=DETECT,gm,sx,sy,tx,ty;
    char text[50];
    void move(int,int,int,int,char[]);
    printf("Enter the text:");
    scanf("%s",text);
    printf("Enter the initial points:");
    scanf("%d%d",&sx,&sy);
    printf("Enter the TARGET points:");
    scanf("%d%d",&tx,&ty);
    initgraph(&gd,&gm,"c:\\turboc3\\bgi");
    outtextxy(sx,sy,text);
    move(sx,sy,tx,ty,text);
    getch();
    closegraph();
}
void move(int sx,int sy,int tx,int ty,char text[50])
{
     int dx=tx-sx,dy=ty-sy,steps,k;
     float xin,yin,x=sx,y=sy;
     getch();
     if(abs(dx)>abs(dy))
       steps=abs(dy);
     else
       steps=abs(dy);
     xin=dx/(float)steps;
     yin=dy/(float)steps;
     for(k=0;k<steps;k++)
     {
```

```
            cleardevice();
            x+=xin;
            y+=yin;
            setcolor(15);
            outtextxy(round(x),round(y),text);
            delay(50);
        }
    }
```

Output:

Enter the initial points:

Enter the TARGET points:

2. Write a program for Generating a Screen with different Shape:

Solution:

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
#include<math.h>
#include<dos.h>
#include<stdlib.h>
int maxx,maxy;
main()
{
 FILE *www;
 int vertex[8];
 int gd,gm,i,t,c,s,x,y;
 char ch;
 gm=0;
 gd=DETECT;
 initgraph(&gd,&gm,"c:\\tc\\BGI");
 maxx=getmaxx();
 maxy=getmaxy();
 setcolor(CYAN);
 rectangle(0,0,maxx,maxy);
 while(1)
 {
  if(kbhit())break;
  www=fopen("c:\\out.txt","w");
  for(i=0;i<100;i++)
  {
   fprintf(www,"%d\n",random(32000));
  }
  fclose(www);
  www=fopen("c:\\out.txt","r");
  while(fscanf(www,"%d%d%d%d%d",&t,&c,&s,&x,&y)==5)
  {
   if(kbhit())goto xx;
```

```
t=t%5;
c=c%16;
s=s%100;
x=x%640;
y=y%480;
setcolor(c);
switch(t)
{
      case 0:
      circle(x,y,s);
      break;
      case 1:
      rectangle(x-35+s,y-20+s,x+35+s,y+20+s);
      case 2:
      ellipse(x,y,0,360,s+20,s);
   case 3:
   bar3d(x-35-s,y-20-s,x+35+s,y+20+s,10,1);
   case 4:
   vertex[0]=x-20-s;
   vertex[1]=y+10+s;
   vertex[2]=x+s;
   vertex[3]=y-10-s;
       vertex[4]=x+20+3*s;
   vertex[5]=y+10+s;
   vertex[6]=vertex[0];
   vertex[7]=vertex[1];
   drawpoly(4,vertex);
   break;
  }
  delay(50);
 }
 fclose(www);
}
xx:
}
```

3. Write a program for Implementing a Snake Game using C graphic's program:

Solution:

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

check();
```

```
end();
win();
int m[500],n[500],con=20;
clock_t start,stop;
 void main(void)
{
int gd=DETECT,gm,ch,maxx,maxy,x=13,y=14,p,q,spd=100,temp,a,i,j,t;
initgraph(&gd,&gm,"C:\\turboc3\\bgi");
setcolor(WHITE);
settextstyle(3,0,6);
outtextxy(200,2," SNAKE 2 ");
settextstyle(6,0,2);
outtextxy(20,80," Use Arrow Keys To Direct The Snake ");
outtextxy(20,140," Avoid The Head Of Snake Not To Hit Any Part Of Snake");
outtextxy(20,160," Pick The Beats Untill You Win The Game ");
outtextxy(20,200," Press 'Esc' Anytime To Exit ");
outtextxy(20,220," Press Any Key To Continue ");
ch=getch();
if(ch==27) exit(0);
cleardevice();
maxx=getmaxx();
maxy=getmaxy();
randomize();
p=random(maxx);
temp=p%13;
p=p-temp;
q=random(maxy);
temp=q%14;
q=q-temp;
 start=clock();
a=0,i=0;
while(1)
{
 setcolor(WHITE);
 setfillstyle(SOLID_FILL,con+5);
 circle(p,q,5);
 floodfill(p,q,WHITE);

  if( kbhit() )
  {
   ch=getch(); if(ch==0) ch=getch();
   if(ch==72&& a!=2) a=1;
   if(ch==80&& a!=1) a=2;
   if(ch==75&& a!=4) a=3;
   if(ch==77&& a!=3) a=4;
    }
     else
    {
```

```
if(ch==27
) break;
}

 if(i<20){
                m[i]=x;
                n[i]=y;
                i++;
          }

               if(i>=20)

               {
                        for(j=con;j>=0;j--){
                                m[1+j]=m[j];
                                n[1+j]=n[j];
                                        }
                m[0]=x;
                n[0]=y;

                setcolor(WHITE);
                setfillstyle(SOLID_FILL,con);
                circle(m[0],n[0],8);
                floodfill(m[0],n[0],WHITE);

                setcolor(WHITE);
                for(j=1;j<con;j++){
               setfillstyle(SOLID_FILL,con+j%3);
                circle(m[j],n[j],5);
                floodfill(m[j],n[j],WHITE);
                                }
  delay(spd);

               setcolor(BLACK);
               setfillstyle(SOLID_FILL,BLACK);
                circle(m[0],n[0],8);
                floodfill(m[0],n[0],BLACK);

                setcolor(BLACK);
               setfillstyle(SOLID_FILL,BLACK);
                circle(m[j],n[j],5);
                floodfill(m[j],n[j],BLACK);

               }
 stop=clock();
 t=(stop-start)/CLK_TCK;
 printf(" TIME %d sec   ",t);
 printf("SCORE %d",con-5);
 check();

if(x==p&&y==q) { con=con+5; if(spd>=5) spd=spd-5; else spd=5;
                                if(con>490) win();
p=random(maxx); temp=p%13;  p=p-temp;
q=random(maxy); temp=q%14;   q=q-temp;
```

```
                                        }
      if(a==1)  y =  y-14; if(y<0) { temp=maxy%14;y=maxy-temp;}
      if(a==2)  y =  y+14; if(y>maxy) y=0;
      if(a==3)  x =  x-13; if(x<0) { temp=maxx%13;x=maxx-temp;}
      if(a==4)  x =  x+13; if(x>maxx) x=0;
      if(a==0){  y = y+14 ;  x=x+13; }
                        }

  }
 check(){
    int a;
    for(a=1;a<con;a++)
 if(m[0]==m[a] && n[0]==n[a]) end();
    else continue;
    getch();
    return(0);

    }
 end()

 {

     int j,i;
    setcolor(WHITE);
    for(i=0;i<5;i++){
    delay(500);
     cleardevice();
     delay(500);
    for(j=0;j<=con;j++){
                        setfillstyle(SOLID_FILL,RED);
                        circle(m[j],n[j],5);
                        floodfill(m[j],n[j],WHITE);
                       }

                      }

    settextstyle(3,0,4);
    outtextxy(150,150,"    GAME OVER ");
    getch();
    exit(0);
    return(0);
    }

win()
{
int j,i;
setcolor(WHITE);
  for(i=0;i<5;i++){
```