**8608113712**

**PROGRAM:**

#include<stdio.h>

#include<conio.h>

#include<dos.h>

#include<math.h>

#include<graphics.h>

void main()

{

float x,y,x1,y1,x2,y2,dx,dy,e;

int i,gd=DETECT,gm;

clrscr();

printf("\n ENTER THE VALUE OF X1:");

scanf("%f",&x1);

printf("\n ENTER THE VALUES OF Y1:");

scanf("%f",&y1);

printf("\n ENTER THE VALUES OF X2 AND Y2:");

scanf("%f%f",&x2,&y2);

initgraph(&gd,&gm," c:\\turboC3\\bgi");

dx=abs(x2-x1);

dy=abs(y2-y1);

x=x1;

y-y1;

e=2\*(dy-dx);

i=1;

do

{

putpixel(x,y,15);

while(e>=0)

{

y=y+1;

e=e-(2\*dx);

}

x=x+1;

e=e+(2\*dy);

i=i+1;

delay(100);

}while(i<=dx);

closegraph();

getch();

}

**OUTPUT:**

ENTER THE VALUE OF X1 : 600

ENTER THE VALUE OF Y1 : 400

ENTER THE VALUE OF X2 AND Y2 : 300 400

**PROGRAM:**

#include <graphics.h>

#include <stdlib.h>

#include <stdio.h>

#include <conio.h>

void draw(int xa,int ya,int xb,int yb);

void main()

{

int xa,ya,xb,yb;

clrscr();

printf("Line DDA algorithm");

printf("\n Enter the value of xa, ya:");

scanf("%d%d",&xa,&ya);

printf("\n Enter the value of xb, yb:");

scanf("%d%d",&xb,&yb);

draw(xa,ya,xb,yb);

}

void draw(int xa,int ya,int xb,int yb)

{

int xin,yin,x,y,dx,dy,steps,k;

int gdriver=DETECT,gmode,errorcode;

initgraph(&gdriver,&gmode, "c:\\turboC3\\bgi")

errorcode=graphresult();

if (errorcode!=grOk)

{

printf("Graphics error: %s\n", grapherrormsg(errorcode));

printf("Press any key to halt:");

getch();

exit(1);

}

dx=xb-xa;

dy=yb-ya;

if(abs(dx)>abs(dy))

{

steps=abs(dx);

}

else

{

steps=abs(dy);

}

xin=dx/steps;

yin=dy/steps;

x=xa;

y=ya;

putpixel(x,y,1);

for(k=1;k<=steps;k++)

{

x=x+xin;

y=y+yin;

putpixel(x,y,1);

}

getch();

closegraph();

}

**OUTPUT:**

Line DDA algorithm

Enter the value of xa, ya:

300

300

Enter the value of xb, yb:

400

400

**PROGRAM:**

#include<stdio.h>

#include<conio.h>

#include<graphics.h>

void main()

{

int gd=DETECT,gm;

int x,y,r;

void fun(int,int,int);

printf(“Enter the midpoints and radius:”);

scanf(“%d%d%d”,&x,&y,&r);

initgraph(&gd,&gm,”c:\\turoboC3\\bgi”);

fun(x,y,r);

getch();

closegraph();

}

void fun(int x1,int y1,int r)

{

int x=0,y=1,p=1-r;

void cliplot(int,int,int,int);

cliplot(x1,y1,x,y);

while(x<y)

{

x++;

if(p<0)

p+=2\*\*+1;

else

{

y--;

p+=2\*(x-y)+1;

}

cliplot(x1,y1,x,y);

}}

void cliplot(int xctr,int yctr,int x,int y)

{

putpixel(xctr+x,yctr+y,1);

putpixel(xctr-x,yctr+y,1);

putpixel(xctr+x,yctr-y,1);

putpixel(xctr-x,yctr-y,1);

putpixel(xctr+y,yctr+x,1);

putpixel(xctr-y,yctr+x,1);

putpixel(xctr+y,yctr-x,1);

putpixel(xctr-y,yctr-x,1);

getch();}

**OUTPUT:**

Enter the midpoints and radius:

100 100 50

**PROGRAM:**

#include<stdio.h>

#include<conio.h>

#include<graphics.h>

#include<math.h>

void main()

{

int gd=DETECT,gm;

int xcenter,ycenter,rx,ry;

int p,x,y,px,py,rx1,ry1,rx2,ry2;

initgraph(&gd,&gm,"c:\\turboC3\\bgi");

printf("\n Enter the radius :");

scanf("%d %d",&rx,&ry);

printf("\n Enter the xcenter and ycenter values :");

scanf("%d %d",&xcenter,&ycenter);

ry1=ry\*ry;

rx1=rx\*rx;

ry2=2\*ry1;

rx2=2\*rx1;

x=0;

y=ry;

plotpoints(xcenter,ycenter,x,y);

p=(ry1-(rx1\*ry)+(0.25\*rx1));

px=0;

py=rx2\*y;

while(px<py)

{

x=x+1;

px=px+ry2;

if(p>=0)

{

y=y-1;

py=py-rx2;

}

if(p<0)

p=p+ry1+px;

else

p=p+ry1+px-y;

plotpoints(xcenter,ycenter,x,y);

p=(ry1\*(x+0.5)\*(x\*0.5)+rx1\*(y-1)\*(y-1)-rx1\*ry1);

while(y>0)

{

y=y-1;

py=py-rx2;

if(p<=0)

{

x=x+1;

px=px+ry2;

}

if(p>0)

p=p+rx1-py;

else

p=p+rx1-py+px;

plotpoints(xcenter,ycenter,x,y);

}

}

getch();

return(0);

}

int plotpoints(int xcenter,int ycenter,int x,int y)

{

putpixel(xcenter+x,ycenter+y,6);

putpixel(xcenter-x,ycenter+y,6);

putpixel(xcenter+x,ycenter-y,6);

putpixel(xcenter-x,ycenter-y,6);

}

**OUTPUT:**

Enter the radius:

10

30

Enter the xcenter and ycenter values:

300

150

**PROGRAM:**

#include<graphics.h>

#include<stdlib.h>

#include<stdio.h>

#include<conio.h>

int x1,x2,x3,y1,y2,y3,t,tx,sx,sy,shx,shy,ch;

float rx1,rx2,rx3,ry1,ry2,ry3;

float ang,theta;

int main(void)

{

int gdriver = DETECT, gmode, errorcode;

initgraph(&gdriver, &gmode,"c:\\tuboC3\\bgi");

errorcode = graphresult();

if(errorcode != grOk)

{

printf("Graphics error: %s\n", grapherrormsg(errorcode));

printf("Press any key to halt:");

getch();

exit(1);

}

else

{

do

{

printf("\n1.Translation\n2.Reflection\n3.Rotation\n4.Scaling\n5.Shearing\n");

printf("\nEnter Your choice");

scanf("%d",&ch);

switch(ch)

{

case 1:

printf("\n Enter all coordinates values :");

scanf("%d %d %d %d %d %d",&x1,&y1,&x2,&y2,&x3,&y3);

printf("\n Before Translation ");

line(x1,y1,x2,y2);

line(x2,y2,x3,y3);

line(x3,y3,x1,y1);

printf("\n Enter the value tranlsation factor :");

scanf("%d",&tx);

printf("\n After Translation\n ");

line(x1+tx,y1,x2+tx,y2);

line(x2+tx,y2,x3+tx,y3);

line(x3+tx,y3,x1+tx,y1);

break;

case 2:

printf("\n Enter all coordinates values :");

scanf("%d %d %d %d %d %d",&x1,&y1,&x2,&y2,&x3,&y3);

printf("\n Before Reflection ");

line(x1,y1,x2,y2);

line(x2,y2,x3,y3);

line(x3,y3,x1,y1);

t=abs(y1-y3);

printf("\n After Reflection ");

line(x1,y1+10+(2\*t),x2,y2+10);

line(x2,y2+10,x3,y3+10);

line(x3,y3+10,x1,y1+10+(2\*t));

break;

case 3:

printf("\n Enter all coordinates values :");

scanf("%d %d %d %d %d %d",&x1,&y1,&x2,&y2,&x3,&y3);

printf("\n Before Rotation ");

line(x1,y1,x2,y2);

line(x2,y2,x3,y3);

line(x3,y3,x1,y1);

printf("\n Enter the rotation angle :");

scanf("%f",&ang);

theta=((ang\*3.14)/180);

rx1=x1\*cos(theta)-y1\*sin(theta);

rx2=x2\*cos(theta)-y2\*sin(theta);

rx3=x3\*cos(theta)-y3\*sin(theta);

ry1=x1\*sin(theta)+y1\*cos(theta);

ry2=x2\*sin(theta)+y2\*cos(theta);

ry3=x3\*sin(theta)+y3\*cos(theta);

printf("\n After Rotation ");

line(rx1,ry1,rx2,ry2);

line(rx2,ry2,rx3,ry3);

line(rx3,ry3,rx1,ry1);

break;

case 4:

printf("\n Enter all coordinates values :");

scanf("%d %d %d %d %d %d",&x1,&y1,&x2,&y2,&x3,&y3);

printf("\n Before Scaling ");

line(x1,y1,x2,y2);

line(x2,y2,x3,y3);

line(x3,y3,x1,y1);

printf("\n Enter the Scale factor :");

scanf("%d %d",&sx,&sy);

printf("\n After Scaling ");

line(x1+sx,y1+sy,x2+sx,y2+sy);

line(x2+sx,y2+sy,x3+sx,y3+sy);

line(x3+sx,y3+sy,x1+sx,y1+sy);

break;

case 5:

printf("\n Enter all coordinates values :");

scanf("%d %d %d %d %d %d",&x1,&y1,&x2,&y2,&x3,&y3);

printf("\n Before Shearing ");

line(x1,y1,x2,y2);

line(x2,y2,x3,y3);

line(x3,y3,x1,y1);

printf("\n Enter 0 for x-axis and 1 for y-axis: ");

scanf("%d",&ch);

if(ch==0)

{

printf("\n Enter the x-SHEAR (^.^) Value: ");

scanf("%d",&shx);

x1=x1+shx\*y1;

x2=x2+shx\*y2;

x3=x3+shx\*y3;

}

else

{

printf("\n Enter the y-SHEAR (^.^) Value: ");

scanf("%d",&shy);

y1=y1+shy\*x1;

y2=y2+shy\*x2;

y3=y3+shy\*x3;

}

printf("\n After Shearing ");

line(x1,y1,x2,y2);

line(x2,y2,x3,y3);

line(x3,y3,x1,y1);

break;

default:

exit(0);

break;

}

}

while(ch!=0);

}

getch();

closegraph();

return 0;

}

**OUTPUT:**

1. Translation

2. Reflection

3. Rotation

4. Scaling

5. Shearing

Enter Your choice 1

Enter all coordinates values 213 236 253 321 256 214

Before Translation

Enter the value translation vector 32

After Translation

1. Translation

2. Reflection

3. Rotation

4. Scaling

5. Shearing

Enter Your choice 2

Enter all coordinates values 213 236 253 321 256 214

Before Reflection

After Reflection

1. Translation

2. Reflection

3. Rotation

4. Scaling

5. Shearing

Enter Your choice 3

Enter all coordinates values 213 236 253 321 256 214

Before Rotation

Enter the rotation angle 20

After Rotation

1. Translation

2. Reflection

3. Rotation

4. Scaling

5. Shearing

Enter Your choice 4

Enter all coordinates values 213 236 253 321 256 214

Before Scaling

Enter the scale factor 10 5

After Scaling

1. Translation

2. Reflection

3. Rotation

4. Scaling

5. Shearing

Enter Your choice 4

Enter all coordinates values 213 236 253 321 256 214

Before Shearing

Enter 0 for x axis and 1 for y axis 0

Enter the x-shear value 1

After Shearing

Before Shearing

Enter 0 for x axis and 1 for y axis 1

Enter the y-shear value 1

After Shearing

**PROGRAM:**

#include<stdio.h>

#include<conio.h>

#include<graphics.h>

#include<math.h>

void 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 coordinates x and y:”);

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);

line(x3.y3,x1,y1);

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 coordinates x and y:200 250 300 350 400 100

**PROGRAM:**

#include <graphics.h>

#include <stdlib.h>

#include <stdio.h>

#include <conio.h>

#include <math.h>

int xa,xb,xc,ya,yb,yc,y1a,y1b,y1c,x1a,x1b,x1c,x2a,x2b,x2c,y2a,y2b,y2c;

int x3a,x3b,x3c,y3a,y3b,y3c,x4a,x4b,x4c,y4a,y4b,y4c,x5a,x5b,x5c,y5a,y5b,y5c;

int tx,shx,t,ch,shy;

float ang,theta,sx,sy;

int main(void)

{

int gdriver = DETECT, gmode, errorcode;

initgraph(&gdriver, &gmode,"c:\\turboC3\\bgi");

printf("\n\t\t\t 2D Composite Transformations");

printf("\n\n Enter all coordinates values :");

scanf("%d %d %d %d %d %d",&xa,&ya,&xb,&yb,&xc,&yc);

printf("\n\n The original Image");

line(xa,ya,xb,yb);

line(xb,yb,xc,yc);

line(xc,yc,xa,ya);

printf("\n\n Enter the value tranlsation factor :");

scanf("%d",&tx);

printf("\n\n After Translation ");

x1a=xa+tx;

x1b=xb+tx;

x1c=xc+tx;

y1a=ya;

y1b=yb;

y1c=yc;

line(x1a,y1a,x1b,y1b);

line(x1b,y1b,x1c,y1c);

line(x1c,y1c,x1a,y1a);

delay(1);

printf("\n\n Next Operation is Rotation");

printf("\n\n Enter the rotation angle :");

scanf("%f",&ang);

theta=((ang\*3.14)/180);

x2a=x1a\*cos(theta)-y1a\*sin(theta);

y2a=x1a\*sin(theta)+y1a\*cos(theta);

x2b=x1b\*cos(theta)-y1b\*sin(theta);

y2b=x1b\*sin(theta)+y1b\*cos(theta);

x2c=x1c\*cos(theta)-y1c\*sin(theta);

y2c=x1c\*sin(theta)+y1c\*cos(theta);

printf("\n\n After Rotation ");

line(x2a,y2a,x2b,y2b);

line(x2b,y2b,x2c,y2c);

line(x2c,y2c,x2a,y2a);

delay(1);

printf("\n\n Next Operation is Scaling");

printf("\n\n Enter the Scale factor :");

scanf("%f %f",&sx,&sy);

x3a=x2a+sx;

y3a=y2a+sy;

x3b=x2b+sx;

y3b=y2b+sy;

x3c=x2c+sx;

y3c=y2c+sy;

printf("\n\n After Scaling ");

line(x3a,y3a,x3b,y3b);

line(x3b,y3b,x3c,y3c);

line(x3c,y3c,x3a,y3a);

delay(1);

printf("\n\n Next Operation is Shearing");

printf("\n\n Enter 1 for x-axis \n 2 for y-axis: ");

scanf("%d",&ch);

if(ch==1) /\* get the shear value\*/

{

printf("\n\n Enter the x-SHEAR (^.^) Value: ");

scanf("%d",&shx);

}

else

{

printf("\n\n Enter the y-SHEAR (^.^) Value: ");

scanf("%d",&shy);

}

if(ch==1)

{

x3a=x3a+shx\*y3a;

y4a=y3a;

x3b=x3a+shx\*y3a;

y4b=y3b;

x3c=x3a+shx\*y3a;

y4c=y3c;

}

else

{

x4a=x3a;

y3a=y3a+shy\*x3a;

x4b=x3b;

y3b=y3b+shy\*x3b;

x4c=x3c;

y3c=y3c+shy\*x3c;

}

printf("\n\n After Shearing ");

line(x3a,y3a,x3b,y3b);

line(x3b,y3b,x3c,y3c);

line(x3c,y3c,x3a,y3a);

delay(1);

printf("\n\n Next Operation is Reflection");

t=abs(y3a-y3c);

x5a=x3a;

x5b=x3b;

x5c=x3c;

y5a=y3a+10+(2\*t);

y5b=y3b+10;

y5c=y3c+10;

printf("\n\n After Reflection ");

line(x5a,y5a,x5b,y5b);

line(x5b,y5b,x5c,y5c);

line(x5c,y5c,x5a,y5a);

getch();

closegraph();

return(0);

}

**OUTPUT:**

2D Composite Transformations

Enter all coordinates values 213 236 253 321 256 214

The original Image

Enter the value translation vector 32

After Translation

Next Operation is Rotation

Enter the rotation angle 20

After Rotation

Next Operation is Scaling

Enter the scale factor 10 5

After Scaling

Next Operation is Shearing

Enter 0 for x axis and 1 for y axis 0

Enter the x-shear value 1

After Shearing

Enter 0 for x axis and 1 for y axis 1

Enter the y-shear value 1

Next Operation is Reflection

After Reflection

**PROGRAM:**

#include<iostream.h>

#include<graphics.h>

#include<conio.h>

#include<stdlib.h>

typedef struct coord

{

int x,y;

char code[4];

}pt;

class sulc

{

public:

WWW.VIDYARTHIPLUS.COM

WWW.VIDYARTHIPLUS.COM V+ TEAM

void drawwindow();

void drawline(pt p1,pt p2,int c1);

pt setcode(pt p);

int visibility(pt p1,pt p2);

pt resetendpt(pt p1,pt p2);

};

void sulc::drawwindow()

{

setcolor(WHITE);

line(150,100,450,100);

line(450,100,450,350);

line(450,350,150,350);

line(150,350,150,100);

}

void sulc::drawline(pt p1,pt p2,int c1)

{

setcolor(c1);

line(p1.x,p1.y,p2.x,p2.y);

}

pt sulc::setcode(pt p)

{

pt ptemp;

if(p.y<100)

ptemp.code[0]='1';

else

ptemp.code[0]='0';

if(p.y>350)

ptemp.code[1]='1';

else

ptemp.code[1]='0';

if(p.y>450)

ptemp.code[2]='1';

else

ptemp.code[2]='0';

if(p.y<150)

ptemp.code[3]='1';

else

ptemp.code[3]='0';

ptemp.x=p.x;

ptemp.y=p.y;

return(ptemp);

}

int sulc::visibility(pt p1,pt p2)

{

int i,flag=0;

for(i=0;i<4;i++)

WWW.VIDYARTHIPLUS.COM

WWW.VIDYARTHIPLUS.COM V+ TEAM

{

if((p1.code[i]!='0')||(p2.code[i]!='0'))

flag=1;

}

if(flag==0)

return(0);

for(i=0;i<4;i++)

{

if((p1.code[i]==p2.code[i])&&(p1.code[i]=='1'))

flag=0;}

if(flag==0)

return(1);

return(2);

}

pt sulc::resetendpt(pt p1,pt p2)

{

pt temp;

int x,y,i;

float m,k;

if(p1.code[3]=='1')

x=150;

if(p1.code[2]=='1')

x=450;

if((p1.code[3]=='1')||(p1.code[2]=='1'))

{

m=(float)(p2.y-p1.y)/(p2.x-p1.x);

k=(p1.y+(m\*(x-p1.x)));

temp.y=k;

temp.x=x;

for(i=0;i<4;i++)

temp.code[i]=p1.code[i];

if(temp.y<=350 &&temp.y>=100)

return(temp);

}

if(p1.code[0]=='1')

y=100;

if(p1.code[1]=='1')

y=350;

if((p1.code[0]=='1')||(p1.code[1]=='1'))

{

m=(float)(p2.y-p1.y)/(p2.x-p1.x);

k=(float)p1.x+(float)(y-p1.y)/m;

temp.x=k;

temp.y=y;

for(i=0;i<4;i++)

temp.code[i]=p1.code[i];

WWW.VIDYARTHIPLUS.COM

WWW.VIDYARTHIPLUS.COM V+ TEAM

if(temp.y<=350 &&temp.y>=100)

return(temp);

}

else

return(p1);

}

void main()

{

int gd=DETECT,gm,v;

sulc c1;

pt p1,p2,ptemp;

initgraph(&gd,&gm,"c:\\turboC3\\bgi");

int x1[10],y1[10],x2[10],y2[10];

cleardevice();

int i,n;

settextstyle(4,0,4);

outtext("cohen sutherland line clipping");

cout<<"\n\n enter the no.of lines:";

cin>>n;

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

{

cout<<"\n\n enter end-point1(x1,y1):";

cin>>x1[i]>>y1[i];

cout<<"\n\n enter end-point2(x2,y2):";

cin>>x2[i]>>y2[i];

}

cleardevice();

settextstyle(0,0,3);

outtext("before clipping");

c1.drawwindow();

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

{

p1.x=x1[i];

p1.y=y1[i];

p2.x=x2[i];

p2.y=y2[i];

c1.drawline(p1,p2,15);

}

getch();

cleardevice();

settextstyle(0,0,3);

outtext("after clipping");

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

{

p2.x=x2[i];

p2.y=y2[i];

WWW.VIDYARTHIPLUS.COM

WWW.VIDYARTHIPLUS.COM V+ TEAM

p1=c1.setcode(p1);

p2=c1.setcode(p2);

v=c1.visibility(p1,p2);

switch(v)

{

case 0:

c1.drawwindow();

c1.drawline(p1,p2,15);

break;

case 1:

c1.drawwindow();

break;

case 2:

p1=c1.resetendpt(p1,p2);

p2=c1.resetendpt(p2,p1);

c1.drawwindow();

c1.drawline(p1,p2,15);

break;

}

}

getch();

closegraph();

}

**OUTPUT:**

Enter the no.of lines: 1

Enter end-point1(x1,y1):30 40

Enter end-point1(x2,y2):300 400

**Before clipping**

**After clipping**

**PROGRAM:**

#include<stdio.h>

#include<conio.h>

#include<graphics.h>

#include<math.h>

void fun(int xm,int ym,int xma,int yma,int x1,int y1,int x2,int y2)

{

rectangle(xm,ym,xma,yma);

rectangle(x1,y1,x2,y2);

line(xm,ym,x1,y1);

line(xm,yma,x1,y2);

line(xma,yma,x2,y2);

line(xma,ym,x2,y1);

getch();

}

void main()

{

int x1,y1,x2,y2,xm,ym,xma,yma,dep,ch,d;

int gd=DETECT,gm; initgraph(&gd,&gm,"c:\\turboC3\\bgi ");

printf("\n enter the TOP-LEFT and BOTTOM-RIGHT CORNER:"); scanf("%d%d%d%d",&xm,&ym,&xma,&yma); printf("\n Enter the depth along z axis:");

scanf("%d",&dep);

cleardevice();

d=dep/2;x1=xm+d;y1=ym-d;

x2=xma+d;

y2=yma-d;

fun(xm,ym,xma,yma,x1,y1,x2,y2);

do

{

cleardevice();

outtextxy(20,20,"MENU");

outtextxy(20,30,"..........");

outtextxy(20,40,"1.PARALLEL PROJECTION");

outtextxy(20,50,"2.PRESPECTIVE PROJECTION");

outtextxy(20,60,"3.EXIT");

outtextxy(20,70,"ENTER THE OPTION");

scanf("%d",&ch);

switch(ch)

{

case 1:

rectangle(xma+100,ym+100,xma+100+dep,yma);

outtextxy(xma+150,(ym+10+yma)/2,"SLIDE SHOW");

rectangle(xm,ym+150,xma,yma+150);

outtextxy(xm+20,(ym+yma)/2+250,"FRONT VIEW");

outtextxy(xm+20,(ym+yma)/2+20,"TO VIEW");

rectangle(xm,ym,xma,xm+dep);

getch();

break;

case 2:

fun(xm/2,ym/2,xma/2,yma/2,x1/2,y1/2,x2/2,y2/2);

break;

}

cleardevice();

}

while(ch==1||ch==2);

closegraph();

}

**OUTPUT:**

ENTER THE TOP-LEFT AND BOTTOM RIGHT CORNER:

50 50 100 100

ENTER THE DEPTH ALONG Z AXIS: 20

MENU:

1. PARALLEL PROJECTION

2. PRESPECTIVE PROJECTION

3. EXIT ENTER UR OPTION: 1

Top view

Front view

Side view

MENU:

1. PARALLEL PROJECTION

2. PRESPECTIVE PROJECTION

3. EXIT ENTER UR OPTION: 2