**PROGRAM 21**

**Write a program to implement 3-D perspective projection.**

#include<stdio.h>

#include<graphics.h>

#define ROUND(a) (int)(a+0.5)

void ddaline(int x1, int y1, int x2, int y2)

{

float xsteps, ysteps, x=x1, y=y1;

int dx = x2-x1;

int dy = y2-y1;

int steps,k=1;

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

steps=abs(dx);

else steps=abs(dy);

xsteps= dx/(float)steps;

ysteps= dy/(float)steps;

putpixel(ROUND(x),ROUND(y),15);

while(k<=steps)

{

x+=xsteps;

y+=ysteps;

putpixel(ROUND(x), ROUND(y),15);

k++;

}

}

void project3d(int x[8],int y[8],int z[8],float xPrp,float yPrp,float zPrp)

{

int xp[8],yp[8],i;

for(i=0; i<8;i++)

{

xp[i]=ROUND(x[i]\*(zPrp/(zPrp-z[i]))- xPrp\*(z[i]/(zPrp-z[i])));

yp[i]=ROUND(y[i]\*(zPrp/(zPrp-z[i]))- yPrp\*(z[i]/(zPrp-z[i])));

}

ddaline(xp[0],yp[0],xp[1],yp[1]);

ddaline(xp[1],yp[1],xp[2],yp[2]);

ddaline(xp[2],yp[2],xp[3],yp[3]);

ddaline(xp[3],yp[3],xp[0],yp[0]);

ddaline(xp[4],yp[4],xp[5],yp[5]);

ddaline(xp[5],yp[5],xp[6],yp[6]);

ddaline(xp[6],yp[6],xp[7],yp[7]);

ddaline(xp[7],yp[7],xp[4],yp[4]);

ddaline(xp[0],yp[0],xp[4],yp[4]);

ddaline(xp[1],yp[1],xp[5],yp[5]);

ddaline(xp[2],yp[2],xp[6],yp[6]);

ddaline(xp[3],yp[3],xp[7],yp[7]);

}

int main()

{

int x[8],y[8],z[8],i;

float xPrp,yPrp,zPrp;

int gdriver = DETECT, gmode, errorcode;

initgraph(&gdriver, &gmode, "..\\");

errorcode = graphresult();

if (errorcode != grOk)

{

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

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

getch();

exit(1);

}

printf("Enter the cube coordinates in sequence\n");

for(i=0;i<8;i++)

scanf("%d %d %d",&x[i],&y[i],&z[i]);

printf("Enter the projection reference point(PRP)\n");

scanf("%f %f %f",&xPrp,&yPrp,&zPrp);

project3d(x,y,z,xPrp,yPrp,zPrp);

getch();

closegraph();

return 0;

}

**OUTPUT 21**



