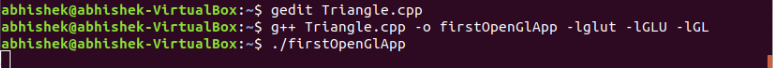
COMPUTER GRAPHICS (OPEN GL)

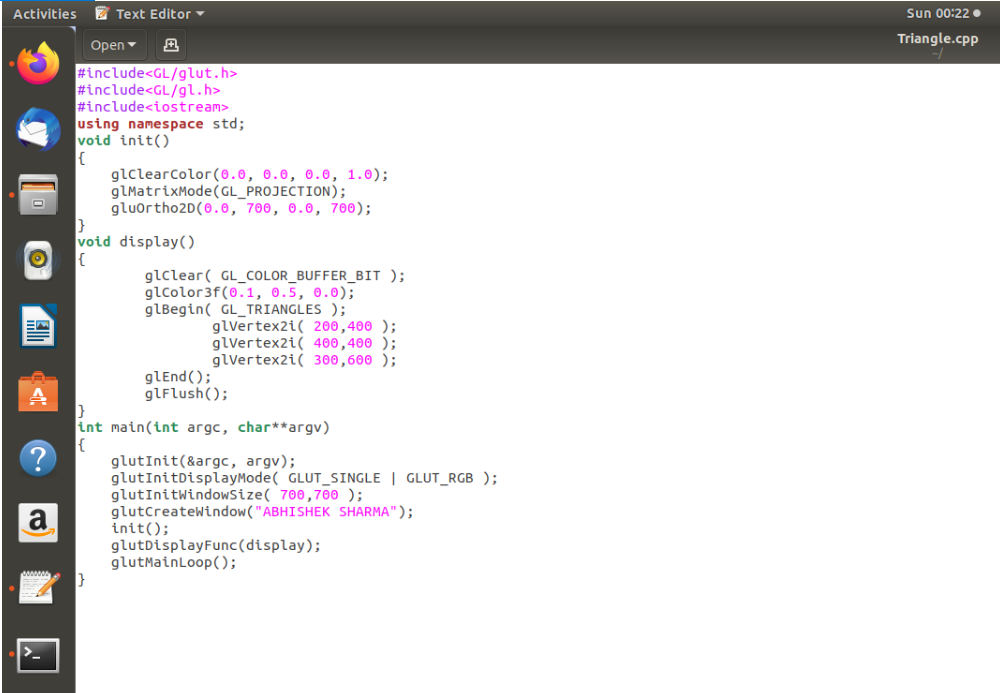
LAB EXPERIMENT FILE

EXPERIMENT 1 (Triangle): -

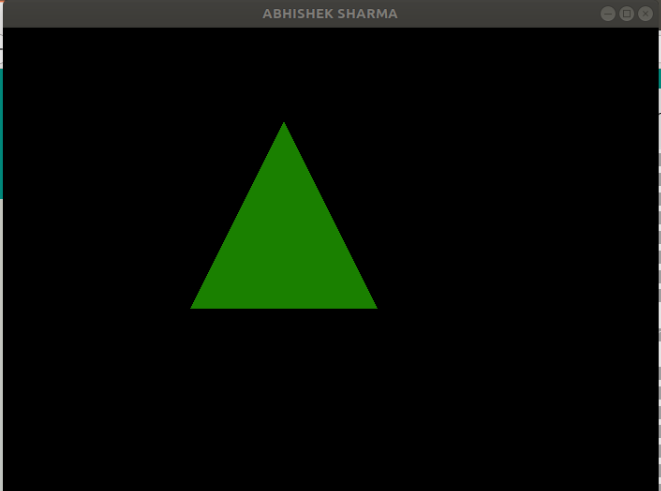
Commands: -



Input code: -

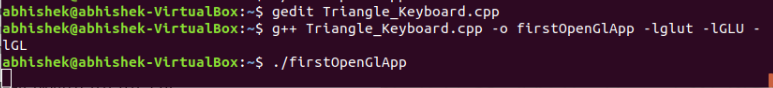


Output: -

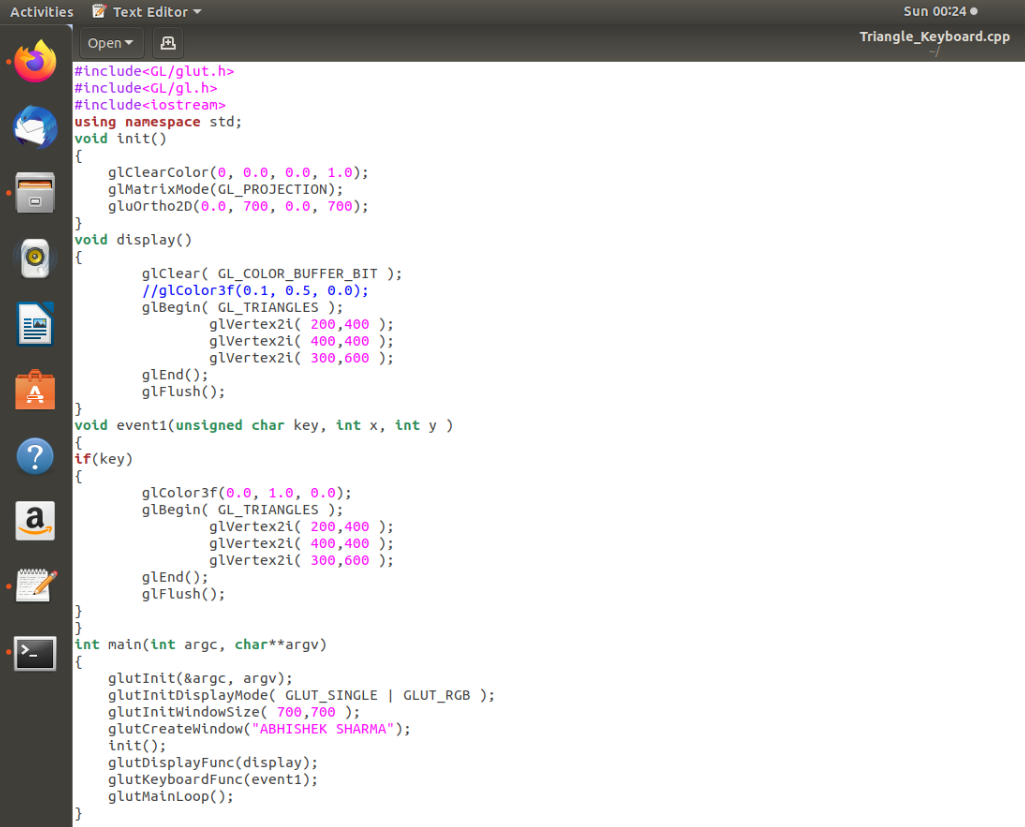
.

EXPERIMENT 2 (Triangle Keyboard): -

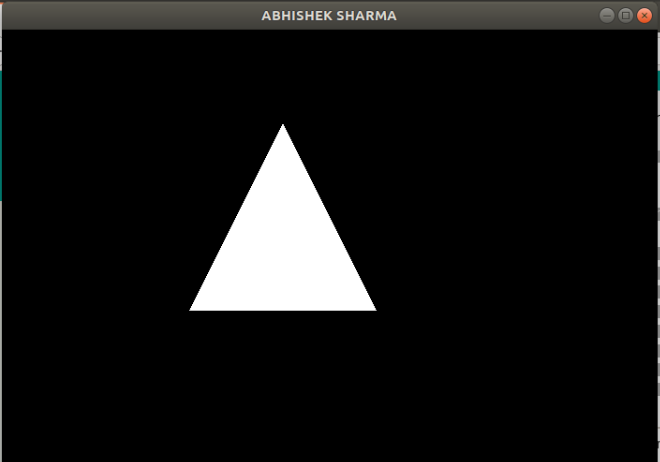
Commands: -

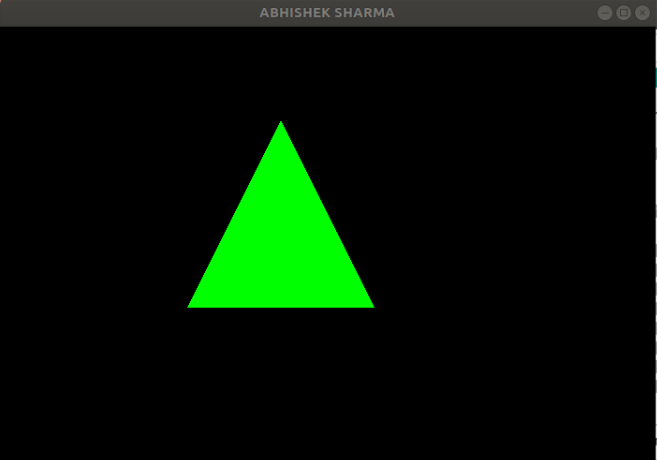


Input code: -



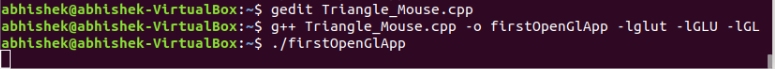
Output: -



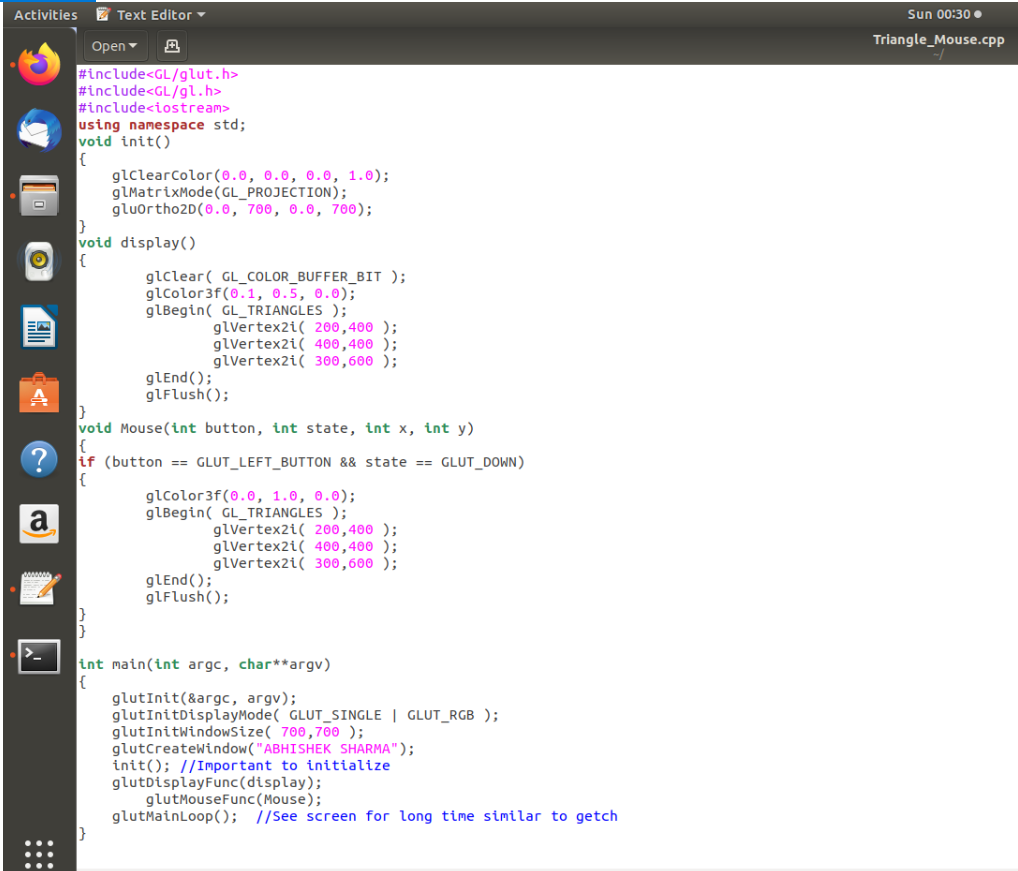


EXPERIMENT 3 (Triangle Mouse): -

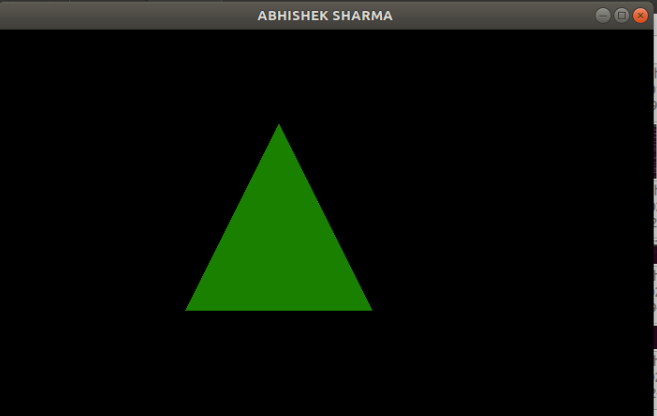
Commands: -

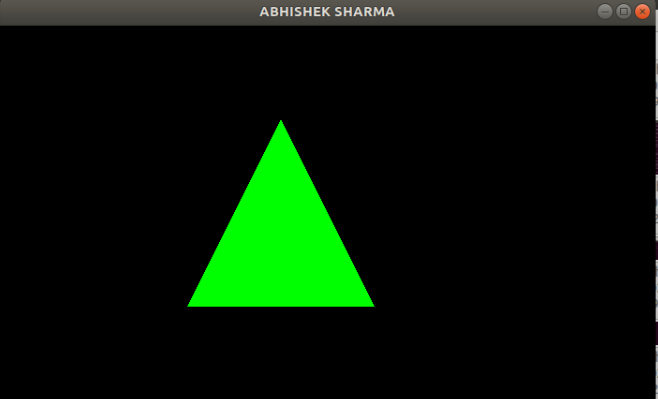


Input code: -

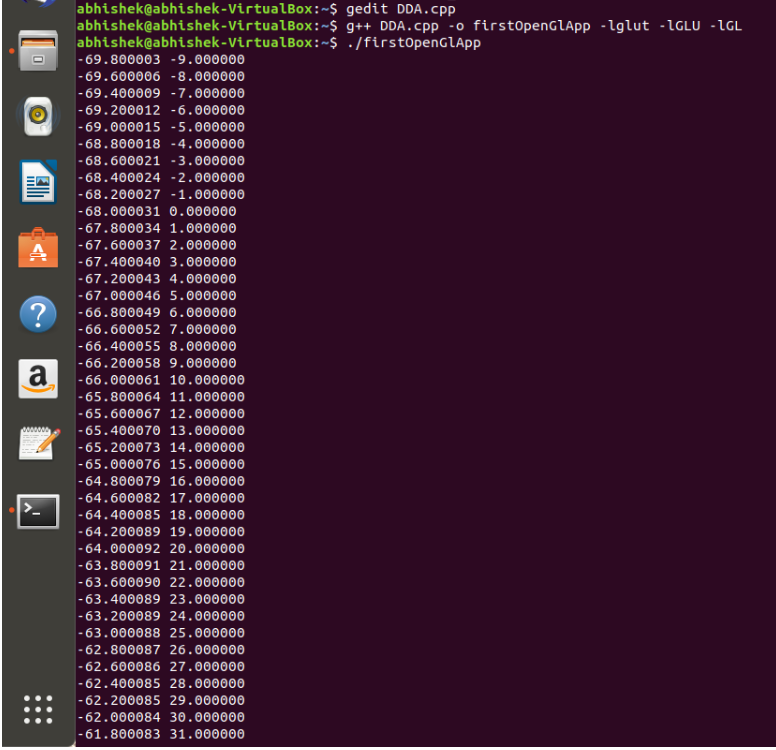


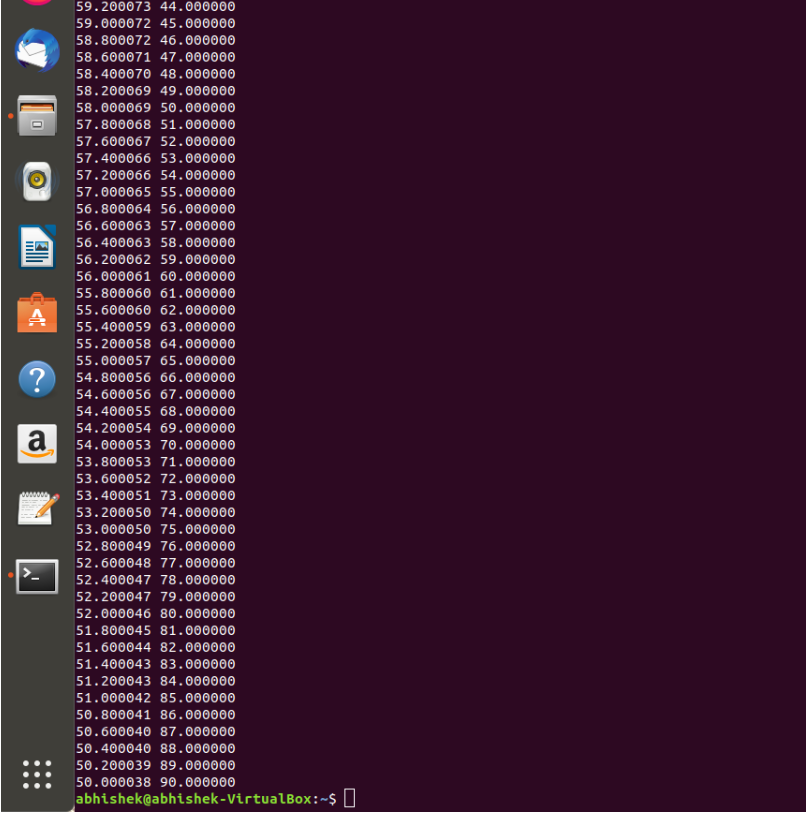
Output: -



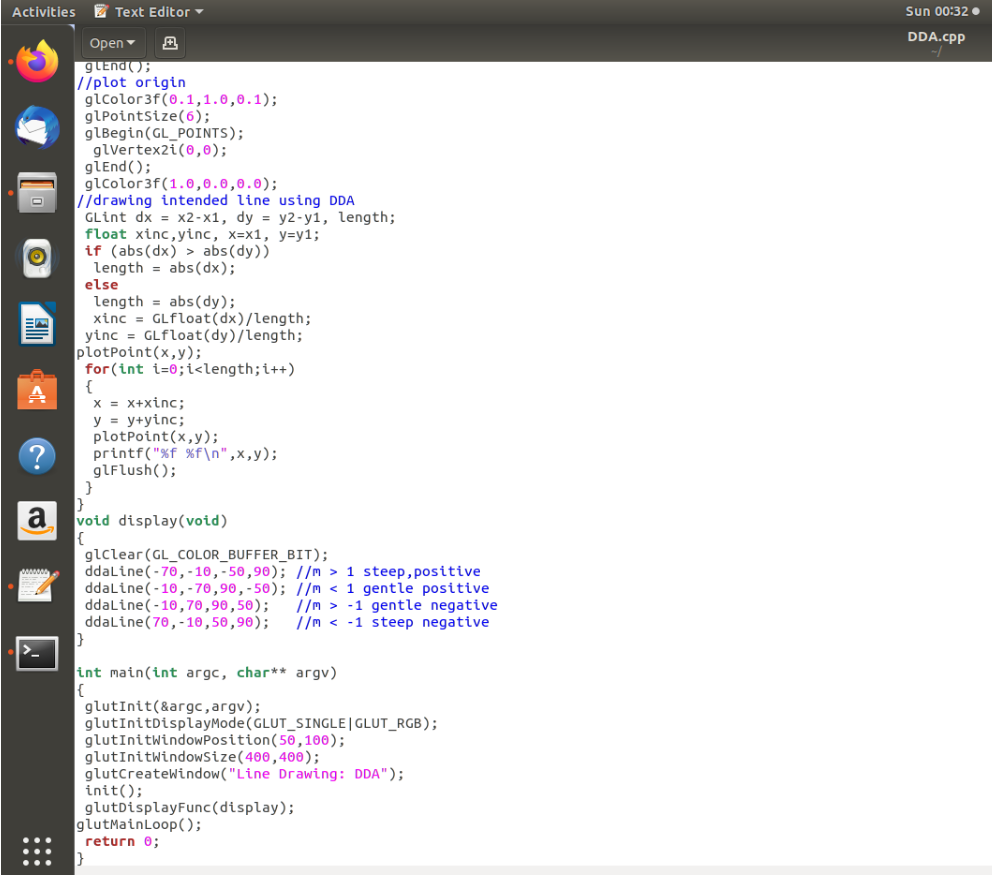


EXPERIMENT 4 (DDA): -

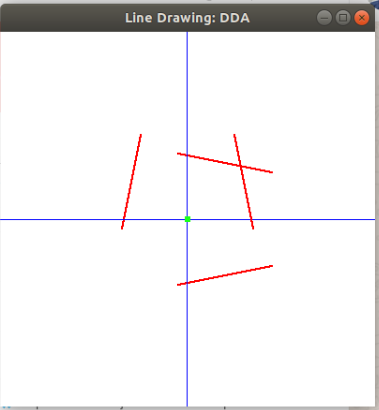
Commands: -



Input code: - 

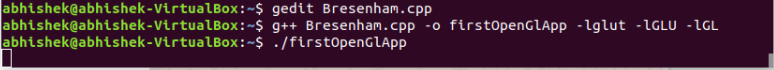


Output: -

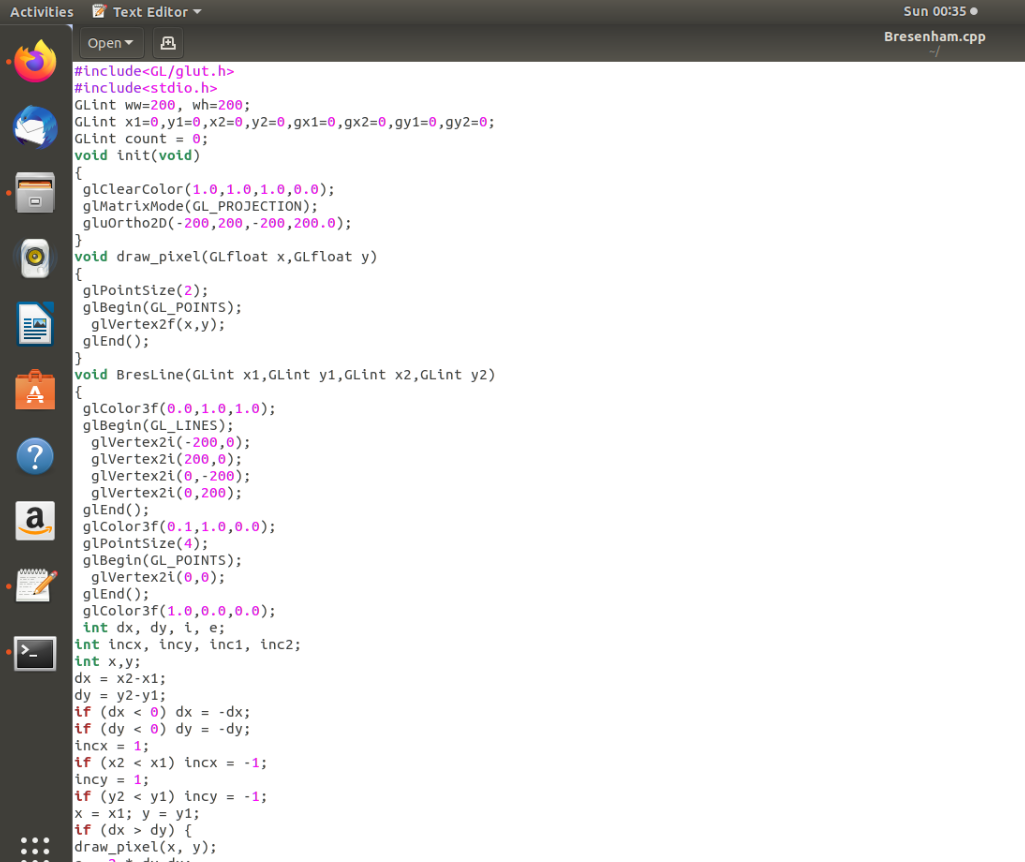


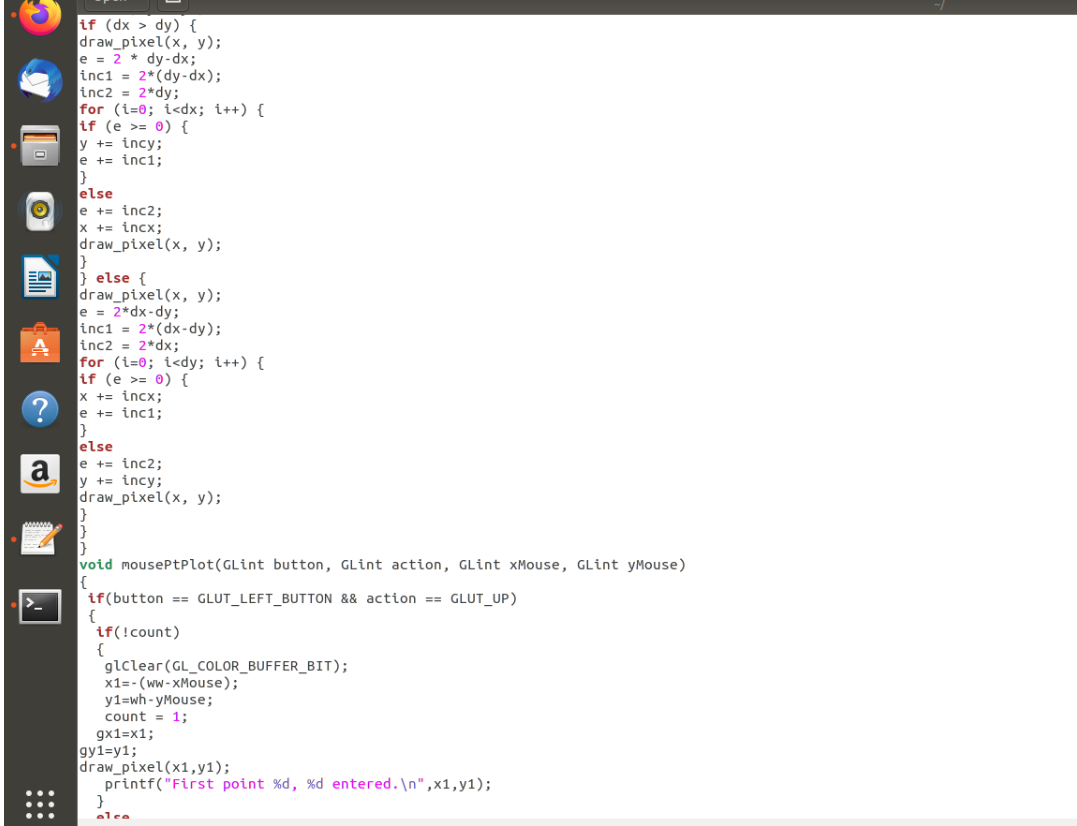
EXPERIMENT 5 (Bresenham): -

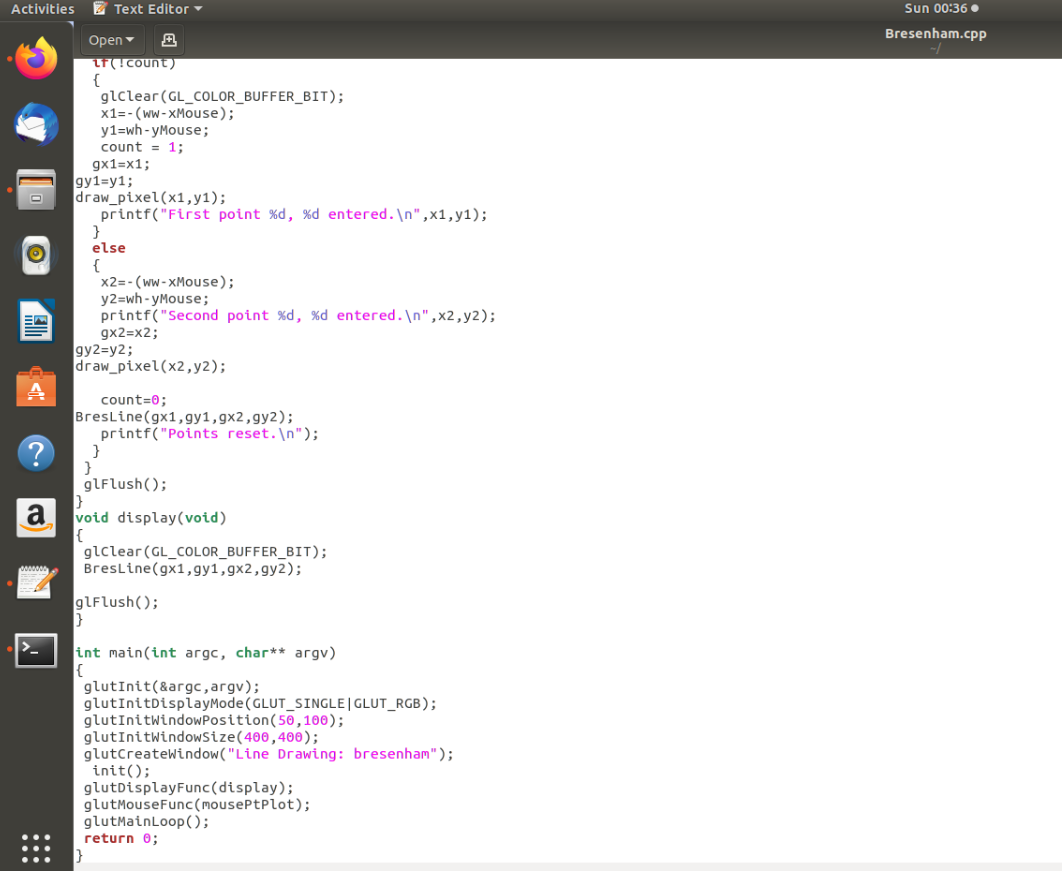
Commands: -



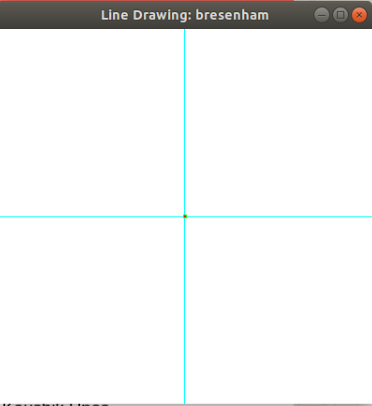
Input code: -





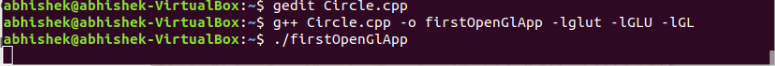


Output: -

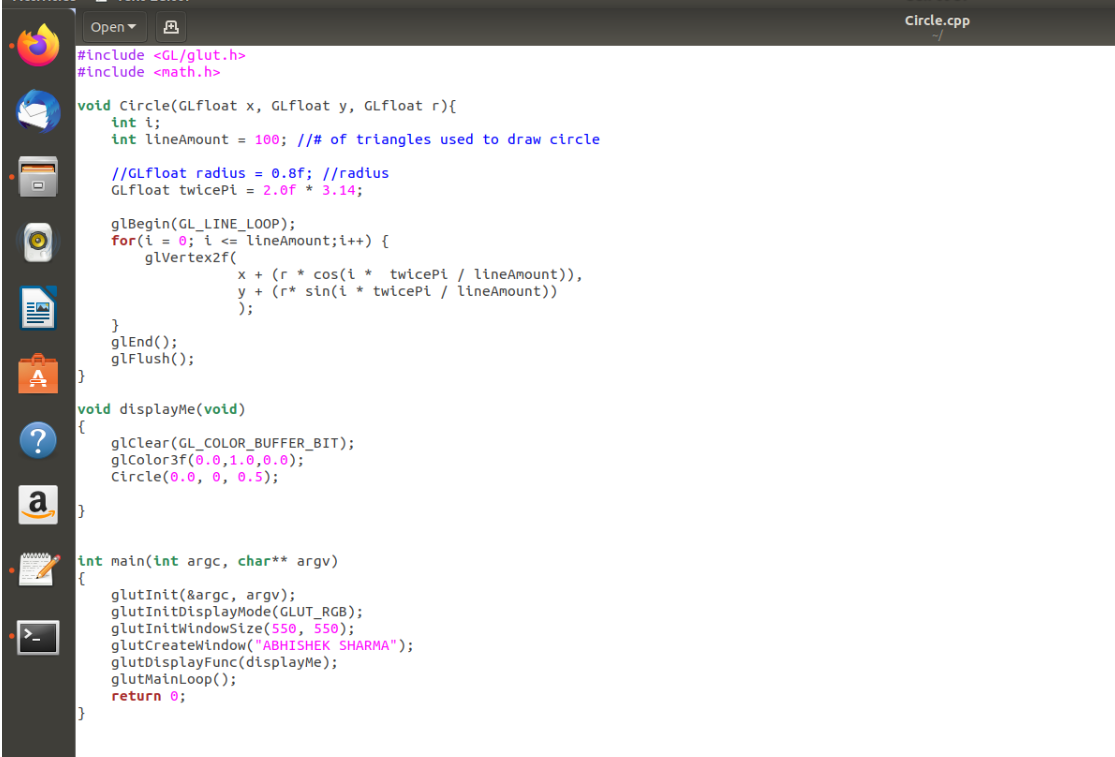


EXPERIMENT 6 (Circle): -

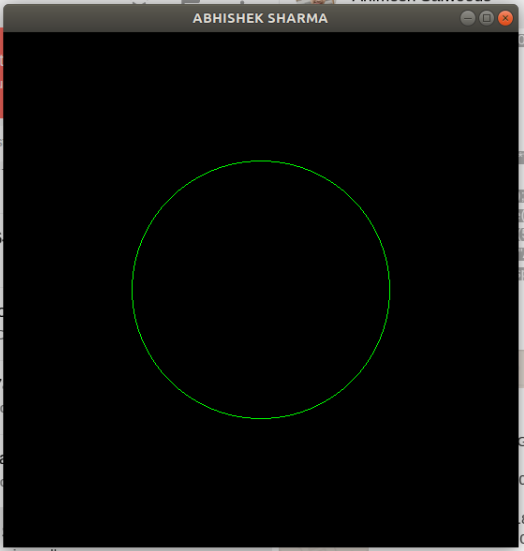
Commands: -



Input code: -

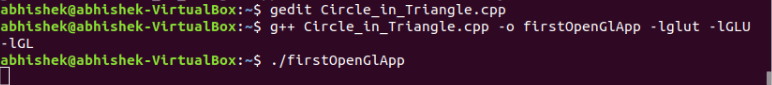


Output: -



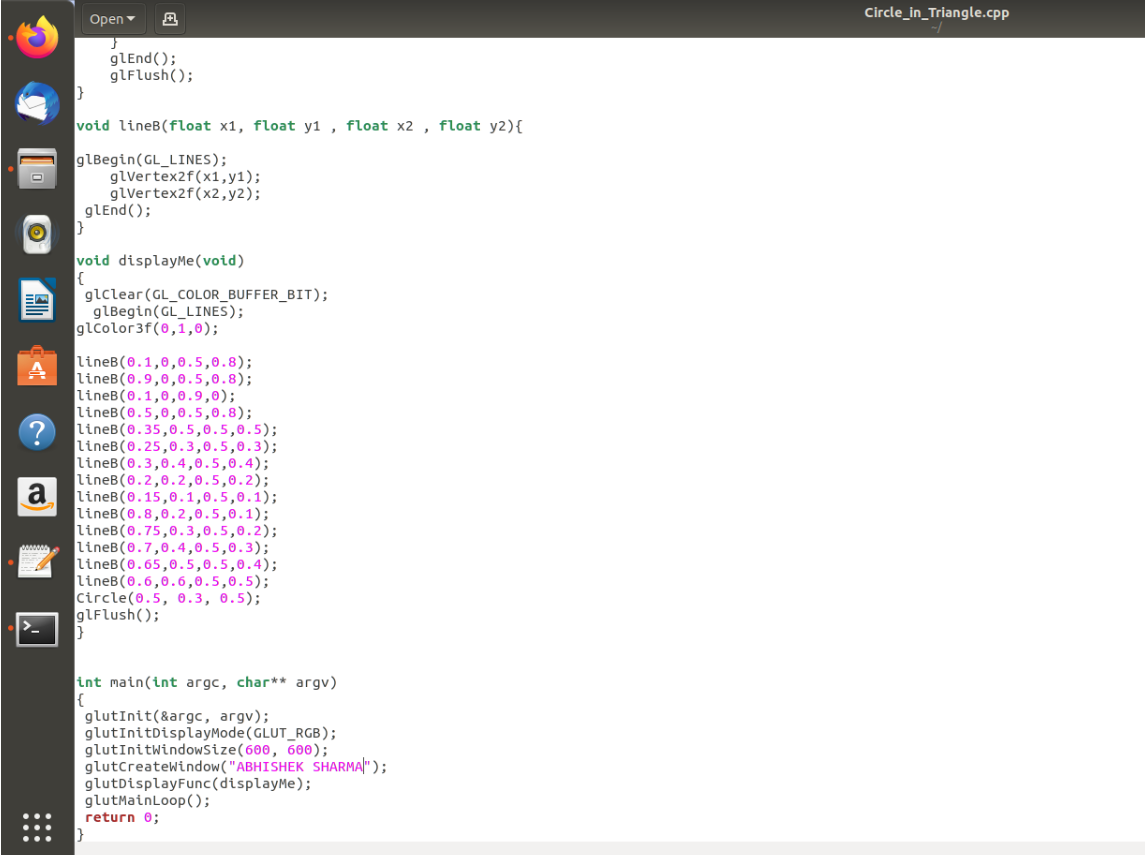
EXPERIMENT 7 (Circle in Triangle): -

Commands: -

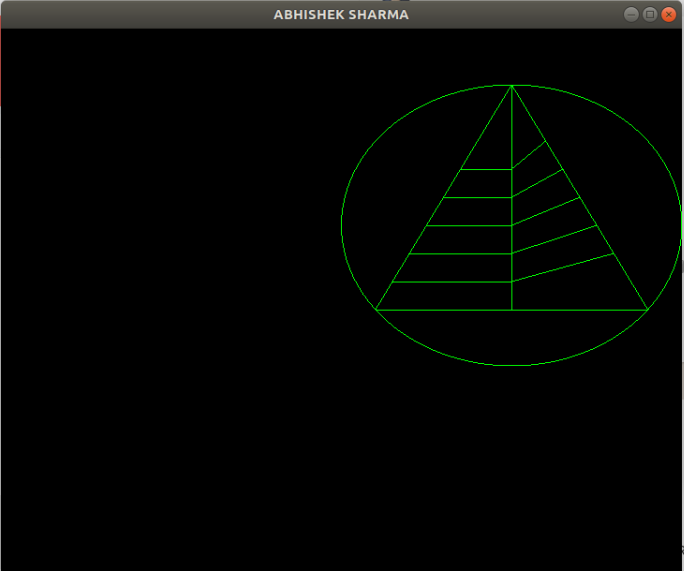


Input code: -



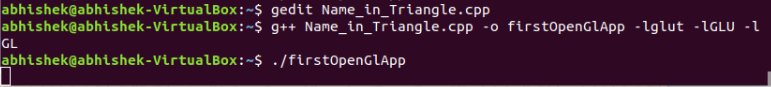


Output: -

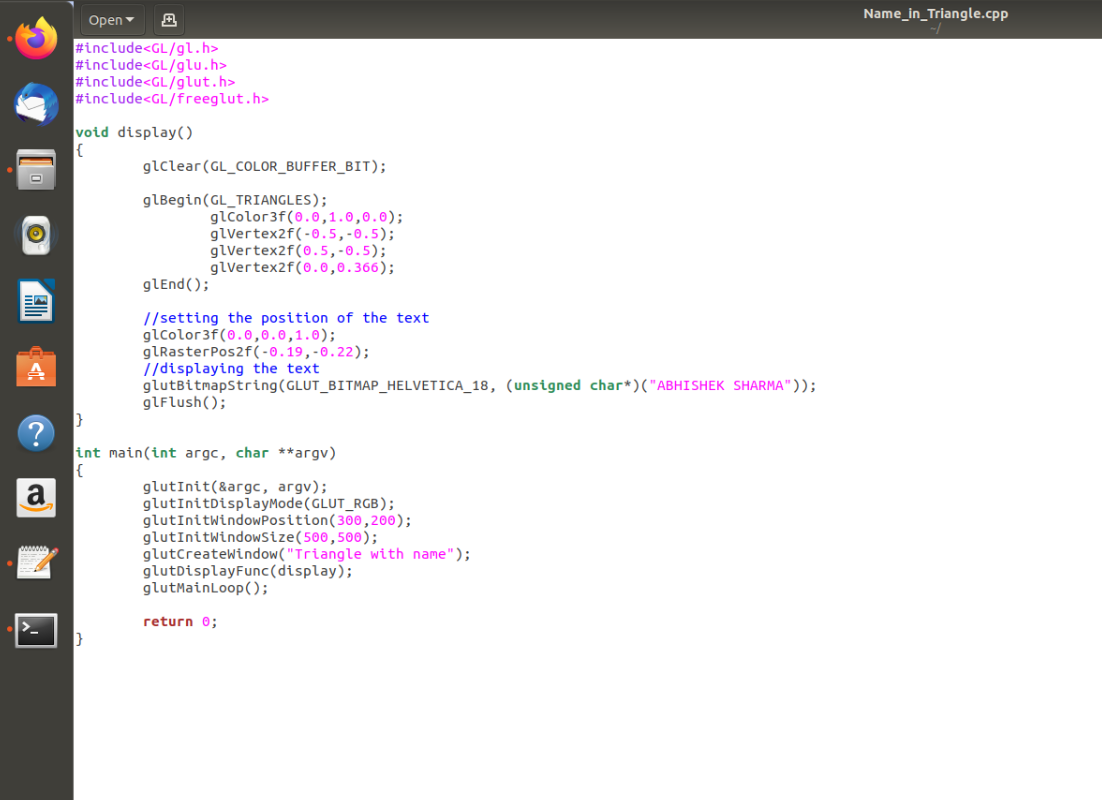


EXPERIMENT 8 (Triangle with Name): -

Commands: -



Input code: -

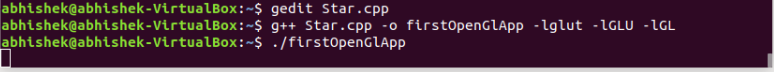


Output: -

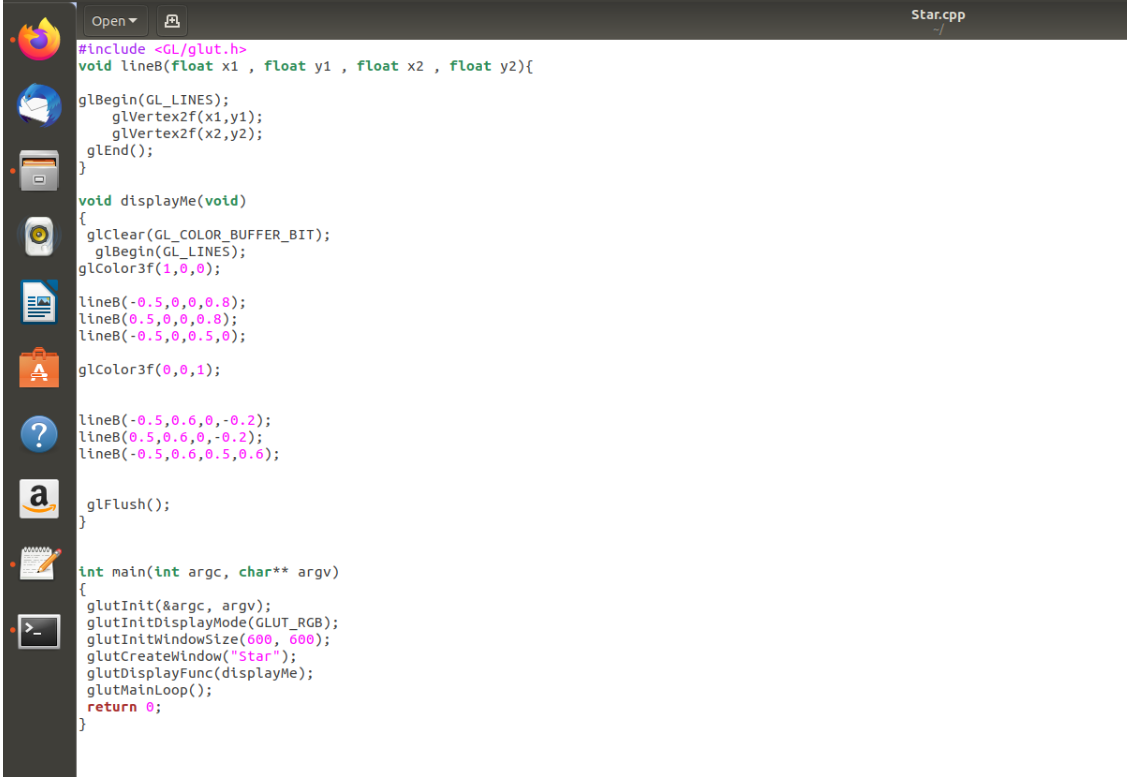


EXPERIMENT 9 (Star): -

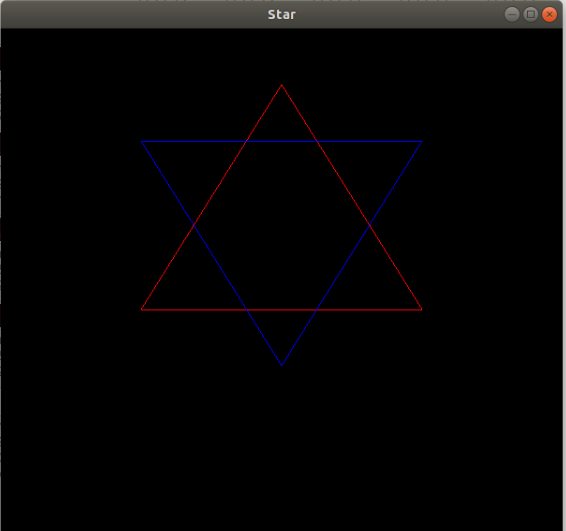
Commands: -



Input code: -

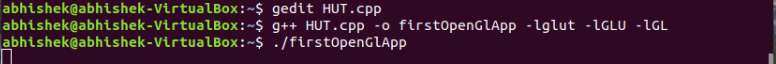


Output: -

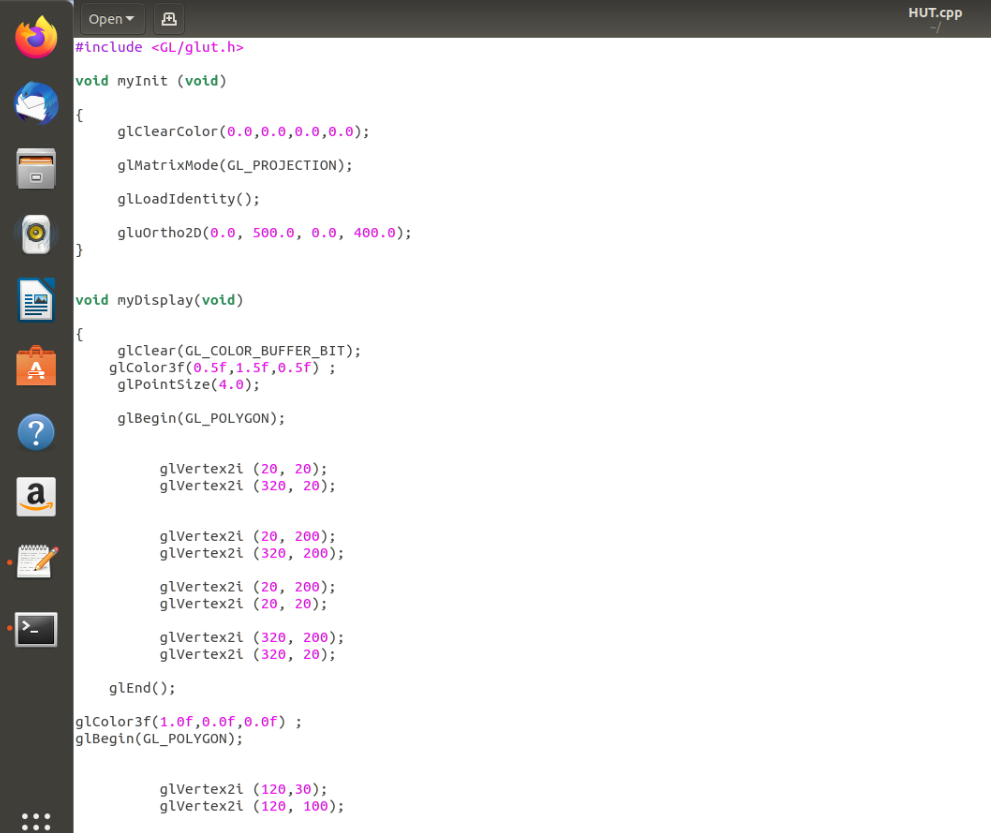
….

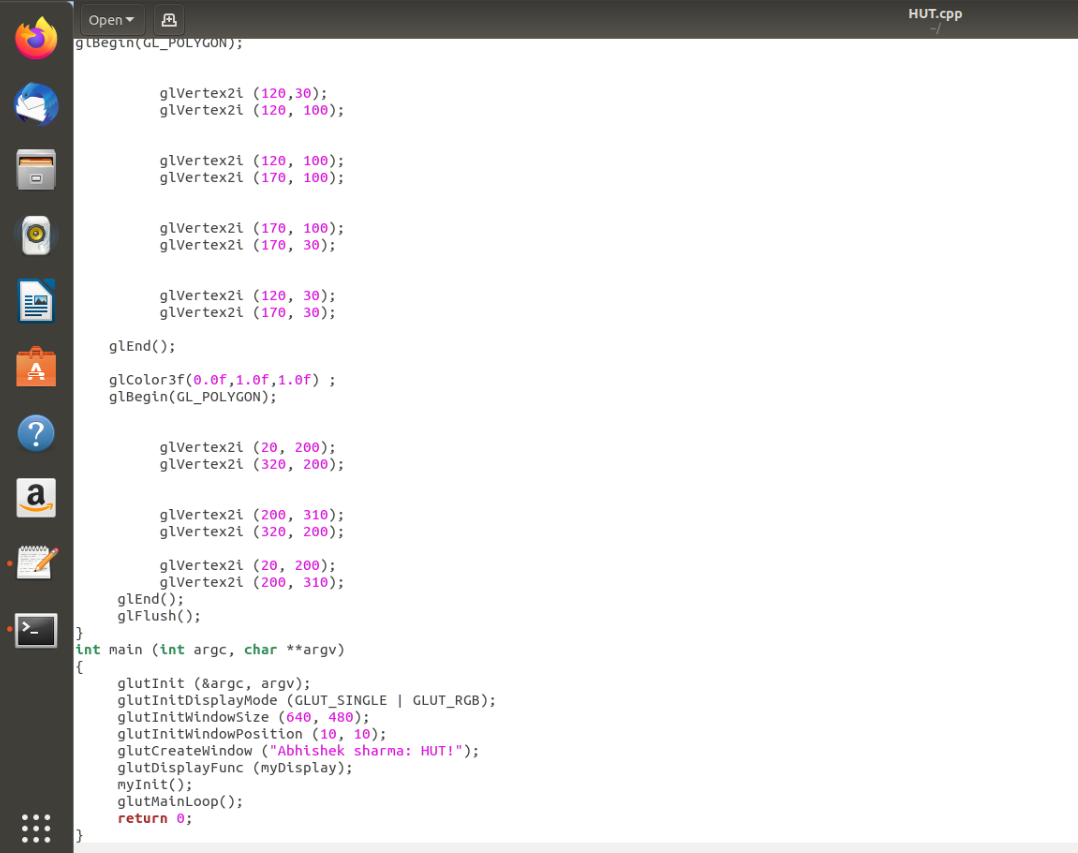
EXPERIMENT 10 (HUT): -

Commands: -

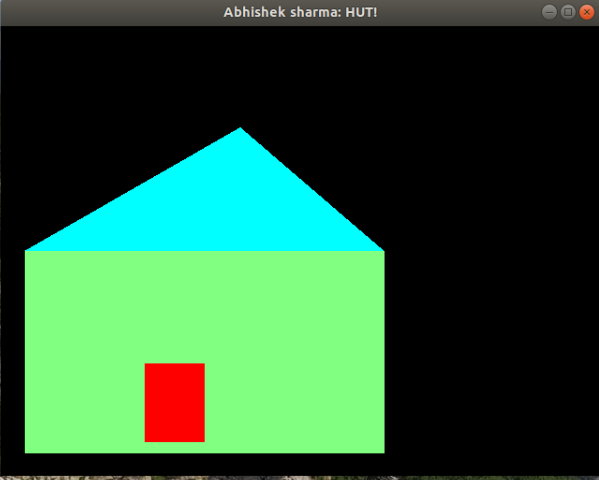


Input code: -





Output: -



**11.Draw a Circle in Triangle in Triangles**

**#include <GL/glut.h>**

**#include <math.h>**

**void line\_join(float x1 , float y1 , float x2 , float y2)**

**glBegin(GL\_LINES);**

**glVertex2f(x1,y1);**

**glVertex2f(x2,y2);**

**}**

**void cirlce()**

**{**

**float theta;**

**glClear(GL\_COLOR\_BUFFER\_BIT);**

**glBegin(GL\_POLYGON);**

**for(int i=0; i<360; i++)**

**{**

**theta = i\*3.142/180;**

**glVertex2f(0.125\*cos(theta),0.125\*sin(theta));**

**}**

**glEnd();**

**glFlush();**

**}**

**void displayMe(void)**

**{**

**glClear(GL\_COLOR\_BUFFER\_BIT);**

**glBegin(GL\_LINES);**

**glColor3f(1.0,1.0,1.0);**

**line\_join(0.0, 1.0, -0.86, -0.50);**

**line\_join(0.0, 1.0, 0.86, -0.50);**

**line\_join(-0.86, -0.50, 0.86, -0.50);**

**glColor3f(0.0,0.0,1.0);**

**line\_join(0.0, -1.0, 0.86, 0.50);**

**line\_join(0.0, -1.0, -0.86, 0.50);**

**line\_join(0.86, 0.50, -0.86, 0.50);**

**glColor3f(1.0,0.0,1.0);**

**line\_join(0.0, 0.5, 0.43, -0.25);**

**line\_join(0.0, 0.5, -0.43, -0.25);**

**line\_join(0.43, -0.25, -0.43, -0.25);**

**glColor3f(0.0,1.0,0.0);**

**line\_join(0.0, -0.250, 0.216, 0.125);**

**line\_join(0.0, -0.250, -0.216, 0.125);**

**line\_join(0.216, 0.125, -0.216, 0.125);**

**glColor3f(1.0,0.0,0.0);**

**cirlce()**

**}**

**int main(int argc, char\*\* argv)**

**{**

**glutInit(&argc, argv);**

**glutInitDisplayMode(GLUT\_RGB);**

**glutInitWindowSize(500, 500);**

**glutCreateWindow("Abhi");**

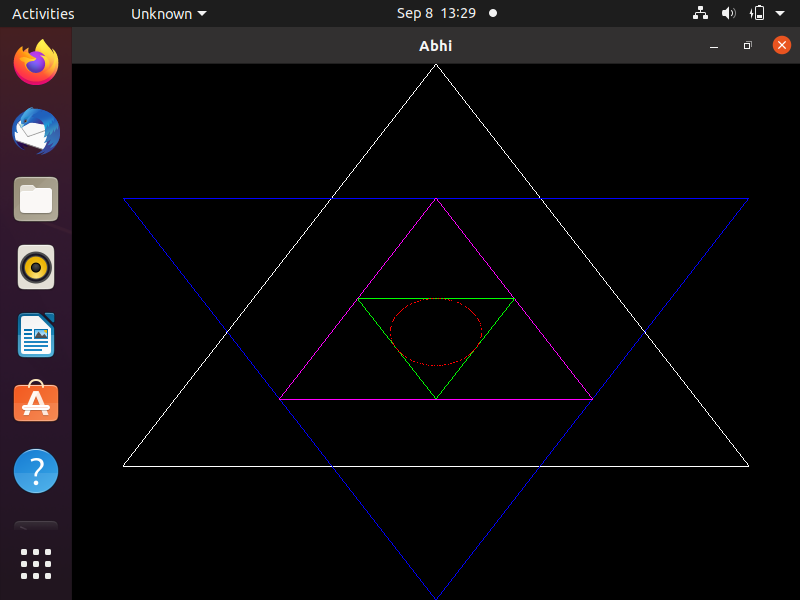
**glutDisplayFunc(displayMe);**

**glutMainLoop();**

**return 0;**

**}**

**OUTPUT**

****

**12.SMILEY**

**#include <GL/glut.h>**

**#include <math.h>**

**void line\_join(float x1 , float y1 , float x2 , float y2){**

**glBegin(GL\_LINES);**

**glVertex2f(x1,y1);**

**glVertex2f(x2,y2);**

**glEnd();**

**glFlush();**

**}**

**void drawHollowCircle(GLfloat x, GLfloat y, GLfloat radius){**

**int i;**

**int lineAmount = 100; //# of triangles used to draw circle**

**//GLfloat radius = 0.8f; //radius**

**GLfloat twicePi = 2.0f \* 3.14;**

**glBegin(GL\_LINE\_LOOP);**

**for(i = 0; i <= lineAmount;i++) {**

**glVertex2f(**

**x + (radius \* cos(i \* twicePi / lineAmount)),**

**y + (radius\* sin(i \* twicePi / lineAmount))**

**);**

**}**

**glEnd();**

**glFlush();**

**}**

**void drawSemiHollowCircle(GLfloat x, GLfloat y, GLfloat radius){**

**int i;**

**int lineAmount = 100;**

**GLfloat twicePi = 2.0f \* 3.14;**

**glBegin(GL\_LINE\_LOOP);**

**for(i = 62; i <= 88;i++) {**

**glVertex2f(**

**x + (radius \* cos(i \* twicePi / lineAmount)),**

**y + (radius\* sin(i \* twicePi / lineAmount))**

**);**

**}**

**glEnd();**

**glFlush();**

**}**

**void displayMe(void)**

**{**

**glClear(GL\_COLOR\_BUFFER\_BIT);**

**glColor3f(1.0,1.0,1.0);**

**drawHollowCircle(-0.2, 0.5, 0.05);**

**drawHollowCircle(-0.2, 0.5, 0.05);**

**drawHollowCircle(0.2, 0.5, 0.05);**

**drawHollowCircle(0.2, 0.5, 0.05);**

**drawHollowCircle(0.0, 0.25, 0.5);**

**line\_join(0.0, 0.15, 0.0, 0.35);**

**drawSemiHollowCircle(0.0, 0.20, 0.25);**

**}**

**int main(int argc, char\*\* argv)**

**{**

**glutInit(&argc, argv);**

**glutInitDisplayMode(GLUT\_RGB);**

**glutInitWindowSize(550, 550);**

**glutCreateWindow("Abhishek");**

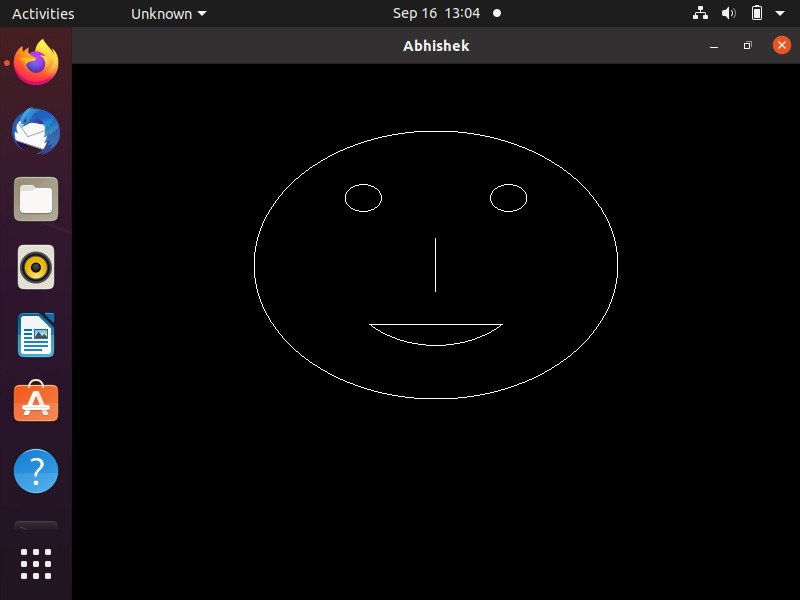
**glutDisplayFunc(displayMe);**

**glutMainLoop();**

**return 0;**

**}**

**OUTPUT:**

****

**13.HISTOGRAM**

**#include<GL/glut.h>**

**#include <cmath>**

**void init()**

**{**

**glClearColor(0,0,0,0);**

**glMatrixMode(GL\_PROJECTION);**

**glOrtho(-100,100,-100,100,20,-20);**

**}**

**void display()**

**{**

**glClear(GL\_COLOR\_BUFFER\_BIT);**

**//glPointSize(2);**

**glBegin(GL\_LINES);**

**glColor3f(1.0,1.0,1.0);**

**glVertex2f(-90,-90);**

**glVertex2f(90,-90);**

**glVertex2f(-90,-90);**

**glVertex2f(-90,90);**

**glEnd();**

**glBegin(GL\_LINES);**

**glColor3f(0.8,0.5,0.0);**

**for(int i=-70;i<-49; i=i+2)**

**{**

**glVertex2f(i,70);**

**glVertex2f(i,-90);**

**}**

**for(int i=70;i>-90;i=i-2)**

**{**

**glVertex2f(-70,i);**

**glVertex2f(-50,i);**

**}**

**glEnd();**

**glBegin(GL\_LINES);**

**glColor3f(0.0,0.5,0.9);**

**for(int i=-40;i<-19; i=i+2)**

**{**

**glVertex2f(i,10);**

**glVertex2f(i,-90);**

**}**

**for(int i=10;i>-90;i=i-2)**

**{**

**glVertex2f(-40,i);**

**glVertex2f(-20,i);**

**}**

**glEnd();**

**glBegin(GL\_LINES);**

**glColor3f(0.5,0.0,0.9);**

**for(int i=-10;i<11; i=i+2)**

**{**

**glVertex2f(i,50);**

**glVertex2f(i,-90);**

**}**

**for(int i=50;i>-90;i=i-2)**

**{**

**glVertex2f(-10,i);**

**glVertex2f(10,i);**

**}**

**glEnd();**

**glBegin(GL\_LINES);**

**glColor3f(0.8,0.0,0.3);**

**for(int i=20;i<41; i=i+2)**

**{**

**glVertex2f(i,80);**

**glVertex2f(i,-90);**

**}**

**for(int i=80;i>-90;i=i-2)**

**{**

**glVertex2f(20,i);**

**glVertex2f(40,i);**

**}**

**glEnd();**

**glBegin(GL\_LINES);**

**glColor3f(0.8,0.8,0.3);**

**for(int i=50;i<71; i=i+2)**

**{**

**glVertex2f(i,30);**

**glVertex2f(i,-90);**

**}**

**for(int i=30;i>-90;i=i-2)**

**{**

**glVertex2f(50,i);**

**glVertex2f(70,i);**

**}**

**glEnd();**

**glFlush();**

**}**

**int main(int argc,char \*\*argv)**

**{**

**glutInit(&argc,argv);**

**glutInitDisplayMode(GLUT\_RGB);**

**glutInitWindowSize(500,500);**

**glutInitWindowPosition(800,100);**

**glutCreateWindow("ABHISHEK SHARMA");**

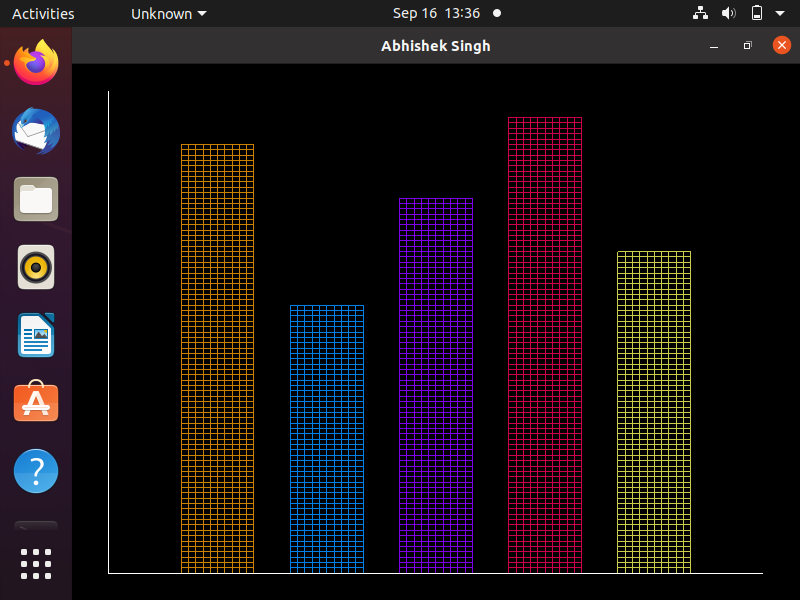
**init();**

**glutDisplayFunc(display);**

**glutMainLoop();**

**}**

**OUTPUT:**

****

**14.Pattern Circles in Pattern Circles**

**#include<GL/glut.h>**

**#include <cmath>**

**void init()**

**{**

**glClearColor(0,0,0,0);**

**glMatrixMode(GL\_PROJECTION);**

**glOrtho(-100,100,-100,100,20,-20);**

**}**

**void display()**

**{**

**glClear(GL\_COLOR\_BUFFER\_BIT);**

**//glPointSize(2);**

**glColor3f(1.0,0.0,1.0);**

**/\*glBegin(GL\_LINES);**

**for(int i=0;i<180;i+10)**

**{**

**float theta1= (i+0.5)\*3.142/180;**

**glVertex2f(8\*cos(theta1),8\*sin(theta1));**

**glVertex2f((8\*cos(theta1)),-(8\*sin(theta1)));**

**}**

**glEnd(); \*/**

**glBegin(GL\_LINES);**

**glColor3f(1.0,0.0,0.0);**

**for(int i=0;i<360;i++)**

**{**

**float theta1= (i+0.5)\*3.142/180;**

**float theta2= (i-0.5)\*3.142/180;**

**glVertex2f(90\*cos(theta1),90\*sin(theta1));**

**glVertex2f(90\*cos(theta2),90\*sin(theta2));**

**}**

**glEnd();**

**float rad, x ,y;**

**glBegin(GL\_LINES);**

**glColor3f(1.0,0.0,1.0);**

**rad = 90;**

**x=0.0f;y=0.0f;**

**while(x<=rad)**

**{**

**y= sqrt((rad\*rad - x\*x));**

**glVertex2f(x,y);**

**glVertex2f(x,-y);**

**glVertex2f(-x,y);**

**glVertex2f(-x,-y);**

**glVertex2f(y,x);**

**glVertex2f(-y,x);**

**glVertex2f(y,-x);**

**glVertex2f(-y,-x);**

**x=x+2;**

**}**

**glEnd();**

**glBegin(GL\_POLYGON);**

**glColor3f(0.0,0.0,0.0);**

**for(int i=0;i<360;i++)**

**{**

**float theta1= (i+0.5)\*3.142/180;**

**glVertex2f(80\*cos(theta1),80\*sin(theta1));**

**}**

**glEnd();**

**glBegin(GL\_LINES);**

**glColor3f(0.0,0.0,1.0);**

**rad = 80;**

**x=0.0f;y=0.0f;**

**while(x<=rad)**

**{**

**y= sqrt((rad\*rad - x\*x));**

**glVertex2f(x,y);**

**glVertex2f(x,-y);**

**glVertex2f(-x,y);**

**glVertex2f(-x,-y);**

**/\***

**glVertex2f(y,x);**

**glVertex2f(-y,x);**

**glVertex2f(y,-x);**

**glVertex2f(-y,-x);**

**\*/**

**x=x+2;**

**}**

**glEnd();**

**glBegin(GL\_LINES);**

**glColor3f(0.0,0.0,1.0);**

**for(int i=0;i<360;i++)**

**{**

**float theta1= (i+0.5)\*3.142/180;**

**float theta2= (i-0.5)\*3.142/180;**

**glVertex2f(80\*cos(theta1),80\*sin(theta1));**

**glVertex2f(80\*cos(theta2),80\*sin(theta2));**

**}**

**glEnd();**

**glBegin(GL\_POLYGON);**

**glColor3f(0.0,0.0,0.0);**

**for(int i=0;i<360;i++)**

**{**

**float theta1= (i+0.5)\*3.142/180;**

**glVertex2f(70\*cos(theta1),70\*sin(theta1));**

**}**

**glEnd();**

**glBegin(GL\_LINES);**

**glColor3f(0.6,0.1,0.2);**

**rad = 70;**

**x=0.0f;y=0.0f;**

**while(x<=rad)**

**{**

**y= sqrt((rad\*rad - x\*x));**

**glVertex2f(x,y);**

**glVertex2f(x,-y);**

**glVertex2f(-x,y);**

**glVertex2f(-x,-y);**

**glVertex2f(y,x);**

**glVertex2f(-y,x);**

**glVertex2f(y,-x);**

**glVertex2f(-y,-x);**

**x=x+2;**

**}**

**glEnd();**

**glBegin(GL\_LINES);**

**glColor3f(1.0,0.0,1.0);**

**for(int i=0;i<360;i++)**

**{**

**float theta1= (i+0.5)\*3.142/180;**

**float theta2= (i-0.5)\*3.142/180;**

**glVertex2f(70\*cos(theta1),70\*sin(theta1));**

**glVertex2f(70\*cos(theta2),70\*sin(theta2));**

**}**

**glEnd();**

**glBegin(GL\_POLYGON);**

**glColor3f(0.0,0.0,0.0);**

**for(int i=0;i<360;i++)**

**{**

**float theta1= (i+0.5)\*3.142/180;**

**glVertex2f(60\*cos(theta1),60\*sin(theta1));**

**}**

**glEnd();**

**glBegin(GL\_LINES);**

**glColor3f(1.0,1.0,1.0);**

**rad = 60;**

**x=0.0f;y=0.0f;**

**while(x<=rad)**

**{**

**y= sqrt((rad\*rad - x\*x));**

**glVertex2f(x,y);**

**glVertex2f(x,-y);**

**glVertex2f(-x,y);**

**glVertex2f(-x,-y);**

**glVertex2f(y,x);**

**glVertex2f(-y,x);**

**glVertex2f(y,-x);**

**glVertex2f(-y,-x);**

**x=x+2;**

**}**

**glEnd();**

**glBegin(GL\_LINES);**

**glColor3f(1.0,1.0,1.0);**

**for(int i=0;i<360;i++)**

**{**

**float theta1= (i+0.5)\*3.142/180;**

**float theta2= (i-0.5)\*3.142/180;**

**glVertex2f(60\*cos(theta1),60\*sin(theta1));**

**glVertex2f(60\*cos(theta2),60\*sin(theta2));**

**}**

**glEnd();**

**glBegin(GL\_POLYGON);**

**glColor3f(0.0,0.0,0.0);**

**for(int i=0;i<360;i++)**

**{**

**float theta1= (i+0.5)\*3.142/180;**

**glVertex2f(50\*cos(theta1),50\*sin(theta1));**

**}**

**glEnd();**

**glBegin(GL\_LINES);**

**glColor3f(0.0,0.9,0.3);**

**rad = 50;**

**x=0.0f;y=0.0f;**

**while(x<=rad)**

**{**

**y= sqrt((rad\*rad - x\*x));**

**/\***

**glVertex2f(x,y);**

**glVertex2f(x,-y);**

**glVertex2f(-x,y);**

**glVertex2f(-x,-y);**

**\*/**

**glVertex2f(y,x);**

**glVertex2f(-y,x);**

**glVertex2f(y,-x);**

**glVertex2f(-y,-x);**

**x=x+2;**

**}**

**glEnd();**

**glBegin(GL\_LINES);**

**glColor3f(0.0,1.0,0.0);**

**for(int i=0;i<360;i++)**

**{**

**float theta1= (i+0.5)\*3.142/180;**

**float theta2= (i-0.5)\*3.142/180;**

**glVertex2f(50\*cos(theta1),50\*sin(theta1));**

**glVertex2f(50\*cos(theta2),50\*sin(theta2));**

**}**

**glEnd();**

**glBegin(GL\_POLYGON);**

**glColor3f(0.0,0.0,0.0);**

**for(int i=0;i<360;i++)**

**{**

**float theta1= (i+0.5)\*3.142/180;**

**glVertex2f(40\*cos(theta1),40\*sin(theta1));**

**}**

**glEnd();**

**glBegin(GL\_LINES);**

**glColor3f(0.0,0.9,0.7);**

**rad = 40;**

**x=0.0f;y=0.0f;**

**while(x<=rad)**

**{**

**y= sqrt((rad\*rad - x\*x));**

**glVertex2f(x,y);**

**glVertex2f(x,-y);**

**glVertex2f(-x,y);**

**glVertex2f(-x,-y);**

**/\***

**glVertex2f(y,x);**

**glVertex2f(-y,x);**

**glVertex2f(y,-x);**

**glVertex2f(-y,-x);**

**\*/**

**x=x+2;**

**}**

**glEnd();**

**glBegin(GL\_LINES);**

**glColor3f(0.0,1.0,0.3);**

**for(int i=0;i<360;i++)**

**{**

**float theta1= (i+0.5)\*3.142/180;**

**float theta2= (i-0.5)\*3.142/180;**

**glVertex2f(40\*cos(theta1),40\*sin(theta1));**

**glVertex2f(40\*cos(theta2),40\*sin(theta2));**

**}**

**glEnd();**

**glBegin(GL\_POLYGON);**

**glColor3f(0.0,0.0,0.0);**

**for(int i=0;i<360;i++)**

**{**

**float theta1= (i+0.5)\*3.142/180;**

**glVertex2f(30\*cos(theta1),30\*sin(theta1));**

**}**

**glEnd();**

**glBegin(GL\_LINES);**

**glColor3f(0.0,0.1,0.9);**

**rad = 30;**

**x=0.0f;y=0.0f;**

**while(x<=rad)**

**{**

**y= sqrt((rad\*rad - x\*x));**

**glVertex2f(x,y);**

**glVertex2f(x,-y);**

**glVertex2f(-x,y);**

**glVertex2f(-x,-y);**

**glVertex2f(y,x);**

**glVertex2f(-y,x);**

**glVertex2f(y,-x);**

**glVertex2f(-y,-x);**

**x=x+2;**

**}**

**glEnd();**

**glBegin(GL\_LINES);**

**glColor3f(0.1,0.3,1.0);**

**for(int i=0;i<360;i++)**

**{**

**float theta1= (i+0.5)\*3.142/180;**

**float theta2= (i-0.5)\*3.142/180;**

**glVertex2f(30\*cos(theta1),30\*sin(theta1));**

**glVertex2f(30\*cos(theta2),30\*sin(theta2));**

**}**

**glEnd();**

**glBegin(GL\_POLYGON);**

**glColor3f(0.0,0.0,0.0);**

**for(int i=0;i<360;i++)**

**{**

**float theta1= (i+0.5)\*3.142/180;**

**glVertex2f(20\*cos(theta1),20\*sin(theta1));**

**}**

**glEnd();**

**glBegin(GL\_LINES);**

**glColor3f(0.5,0.5,0.0);**

**rad = 20;**

**x=0.0f;y=0.0f;**

**while(x<=rad)**

**{**

**y= sqrt((rad\*rad - x\*x));**

**glVertex2f(x,y);**

**glVertex2f(x,-y);**

**glVertex2f(-x,y);**

**glVertex2f(-x,-y);**

**glVertex2f(y,x);**

**glVertex2f(-y,x);**

**glVertex2f(y,-x);**

**glVertex2f(-y,-x);**

**x=x+2;**

**}**

**glEnd();**

**glBegin(GL\_LINES);**

**glColor3f(1.0,0.6,0.0);**

**for(int i=0;i<360;i++)**

**{**

**float theta1= (i+0.5)\*3.142/180;**

**float theta2= (i-0.5)\*3.142/180;**

**glVertex2f(20\*cos(theta1),20\*sin(theta1));**

**glVertex2f(20\*cos(theta2),20\*sin(theta2));**

**}**

**glEnd();**

**glBegin(GL\_POLYGON);**

**glColor3f(0.0,0.0,0.0);**

**for(int i=0;i<360;i++)**

**{**

**float theta1= (i+0.5)\*3.142/180;**

**glVertex2f(10\*cos(theta1),10\*sin(theta1));**

**}**

**glEnd();**

**glBegin(GL\_LINES);**

**glColor3f(0.9,0.9,0.3);**

**rad = 10;**

**x=0.0f;y=0.0f;**

**while(x<=rad)**

**{**

**y= sqrt((rad\*rad - x\*x));**

**glVertex2f(x,y);**

**glVertex2f(x,-y);**

**glVertex2f(-x,y);**

**glVertex2f(-x,-y);**

**glVertex2f(y,x);**

**glVertex2f(-y,x);**

**glVertex2f(y,-x);**

**glVertex2f(-y,-x);**

**x=x+2;**

**}**

**glEnd();**

**glBegin(GL\_LINES);**

**glColor3f(1.0,1.0,0.0);**

**for(int i=0;i<360;i++)**

**{**

**float theta1= (i+0.5)\*3.142/180;**

**float theta2= (i-0.5)\*3.142/180;**

**glVertex2f(10\*cos(theta1),10\*sin(theta1));**

**glVertex2f(10\*cos(theta2),10\*sin(theta2));**

**}**

**glEnd();**

**glFlush();**

**}**

**int main(int argc,char \*\*argv)**

**{**

**glutInit(&argc,argv);**

**glutInitDisplayMode(GLUT\_RGB);**

**glutInitWindowSize(500,500);**

**glutInitWindowPosition(800,100);**

**glutCreateWindow("ABHISHEK SHARMA");**

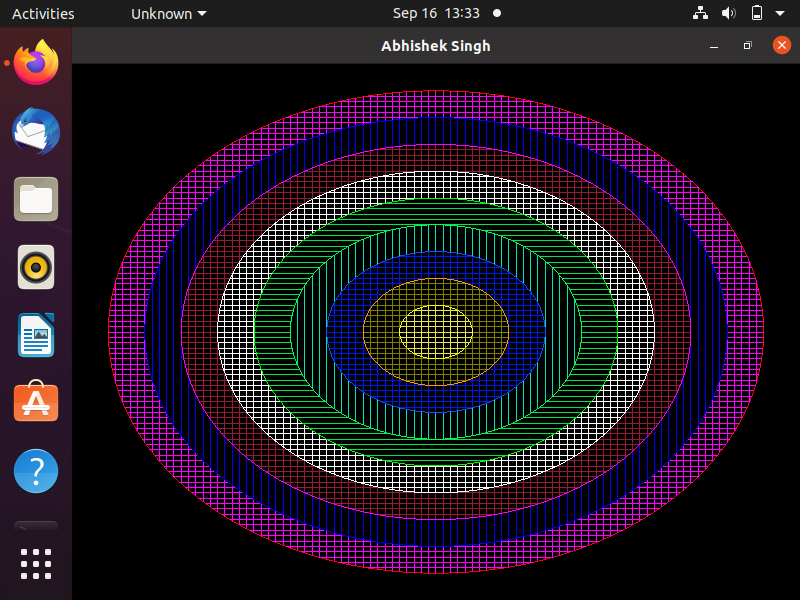
**init();**

**glutDisplayFunc(display);**

**glutMainLoop();**

**}**

**OUTPUT:**

****

**15.Car Animation**

**#include <GL/glut.h>**

**#include <stdlib.h>**

**#include <math.h>**

**GLint b=300;**

**float counter=600.0;**

**void initOpenGl()**

**{**

**glClearColor(0.5,0.5,0.5,1); //Background Color**

**glMatrixMode(GL\_PROJECTION);**

**glLoadIdentity();**

**gluOrtho2D(0,700,0,500);**

**glMatrixMode(GL\_MODELVIEW);**

**}**

**void wheel(int x,int y)**

**{**

**float th;**

**glBegin(GL\_POLYGON);**

**glColor3f(0,0,0);**

**for(int i=0;i<360;i++)**

**{**

**th=i\*(3.1416/180);**

**glVertex2f(x+20\*cos(th),y+20\*sin(th));**

**}**

**glEnd();**

**}**

**void car()**

**{**

**//Bottom Part**

**glLoadIdentity();**

**counter=counter-2;**

**glTranslated(counter,100,0.0);**

**//glScaled(0.1,0.1,0.0);**

**glBegin(GL\_POLYGON);**

**glVertex2f(100,100);**

**glVertex2f(100,160);**

**glVertex2f(450,160);**

**glVertex2f(450,100);**

**//Top Part**

**glBegin(GL\_POLYGON);**

**glVertex2f(150,160);**

**glVertex2f(200,200);**

**glVertex2f(400,200);**

**glVertex2f(450,160);**

**glEnd();**

**wheel(200,100);**

**wheel(380,100);**

**}**

**void display()**

**{**

**glClear(GL\_COLOR\_BUFFER\_BIT);**

**//Push and pop matrix for separating circle object from Background**

**glColor3f(1.0,1.0,1.0);**

**car();**

**glutSwapBuffers();**

**glFlush();**

**}**

**int main(int argc, char \*\*argv)**

**{**

**glutInit(&argc,argv);**

**glutInitDisplayMode(GLUT\_DOUBLE|GLUT\_RGBA|GLUT\_DEPTH);**

**glutInitWindowSize(700,500);**

**glutInitWindowPosition(0,0);**

**glutCreateWindow("Abhishek Sharma");**

**initOpenGl();**

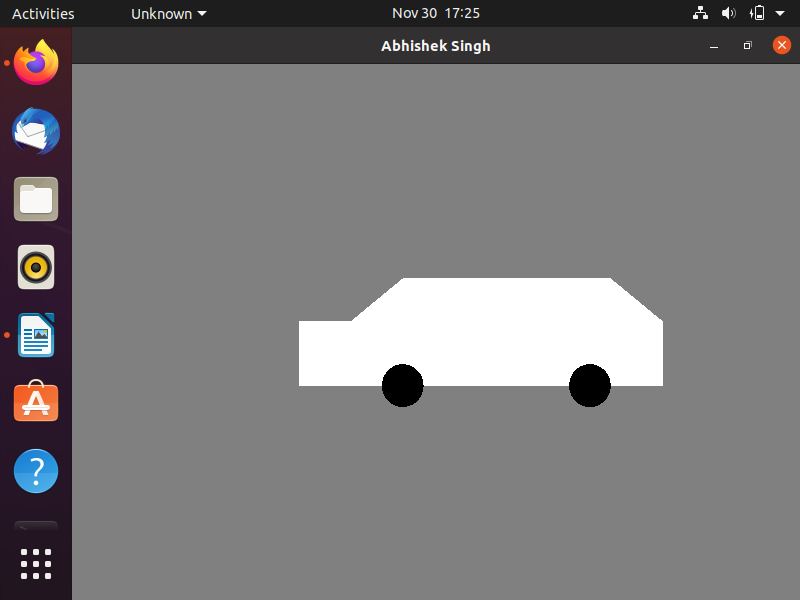
**glutDisplayFunc(display);**

**glutIdleFunc(display);**

**glutMainLoop();**

**return 0;**

**}**

**Output:- **

**16.Clock**

**#include <GL/glut.h>**

**#include <math.h>**

**#include <time.h>**

**const GLfloat tam\_x = 50.0f;**

**const GLfloat tam\_y = 50.0f;**

**const GLint sy = 30;**

**const GLint my = 25;**

**const GLint hy = 20;**

**int hour;**

**int min;**

**int sec;**

**void circulo(GLfloat xc, GLfloat yc, GLfloat raio, bool fill)**

**{**

**const GLfloat c = 3.14169f / 180.0f;**

**GLint i;**

**glBegin(fill ? GL\_TRIANGLE\_FAN : GL\_LINE\_LOOP);**

**for (i = 0; i <= 360; i += 2)**

**{**

**float a = i \* c;**

**glVertex2f(xc + sin(a) \* raio, yc + cos(a) \* raio);**

**}**

**glEnd();**

**}**

**void Display(void)**

**{**

**glClear(GL\_COLOR\_BUFFER\_BIT);**

**glColor3f(1, 0, 1);**

**circulo(0, 0, tam\_x, true);**

**glColor3f(0.0f, 0.0f, 0.0f);**

**circulo(0, 0, tam\_x, false);**

**float anguloS = sec \* 6;**

**glRotatef(-anguloS, 0.0f, 0.0f, 1.0f);**

**glBegin(GL\_LINES);**

**glColor3f(0.0f, 1.0f, 0.0f);**

**glVertex2i(0,0);**

**glVertex2i(0,sy);**

**glEnd();**

**glLoadIdentity();**

**float anguloM = min \* 6;**

**glRotatef(-anguloM, 0.0f, 0.0f, 1.0f);**

**glLineWidth(5);**

**glBegin(GL\_LINES);**

**glColor3f(1.0f, 0.0f, 0.0f);**

**glVertex2i(0,0);**

**glVertex2i(0,my);**

**glEnd();**

**glLoadIdentity();**

**float anguloH = (hour + min/60.0) \* 30;**

**glRotatef(-anguloH, 0.0f, 0.0f, 1.0f);**

**glBegin(GL\_LINES);**

**glColor3f(1, 1, 1);**

**glVertex2i(0,0);**

**glVertex2i(0,hy);**

**glEnd();**

**glLoadIdentity();**

**glFlush();**

**}**

**void resize(GLsizei latitude, GLsizei altitude)**

**{**

**glViewport(0, 0, latitude, altitude);**

**glMatrixMode(GL\_PROJECTION);**

**glLoadIdentity();**

**if (latitude <= altitude)**

**gluOrtho2D(-tam\_x, tam\_x, -tam\_y \* altitude / latitude, tam\_y \* altitude / latitude);**

**else**

**gluOrtho2D(-tam\_x \* latitude / altitude, tam\_x \* latitude / altitude, -tam\_y, tam\_y);**

**glMatrixMode(GL\_MODELVIEW);**

**glLoadIdentity();**

**}**

**void move(int n)**

**{**

**time\_t agora = time(0);**

**struct tm \*datahour = localtime(&agora);**

**hour = datahour->tm\_hour;**

**min = datahour->tm\_min;**

**sec = datahour->tm\_sec;**

**glutPostRedisplay();**

**glutTimerFunc(1000, move, 0);**

**}**

**void initialize (void)**

**{**

**glClearColor(1.0f, 1.0f, 1.0f, 1.0f);**

**}**

**int main(int argc, char\*\* argv) {**

**glutInit(&argc, argv);**

**glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);**

**glutInitWindowSize(400,400);**

**glutInitWindowPosition(10,10);**

**glutCreateWindow("Abhishek Sharma");**

**glutDisplayFunc(Display);**

**glutReshapeFunc(resize);**

**glutTimerFunc(1000, move, 0);**

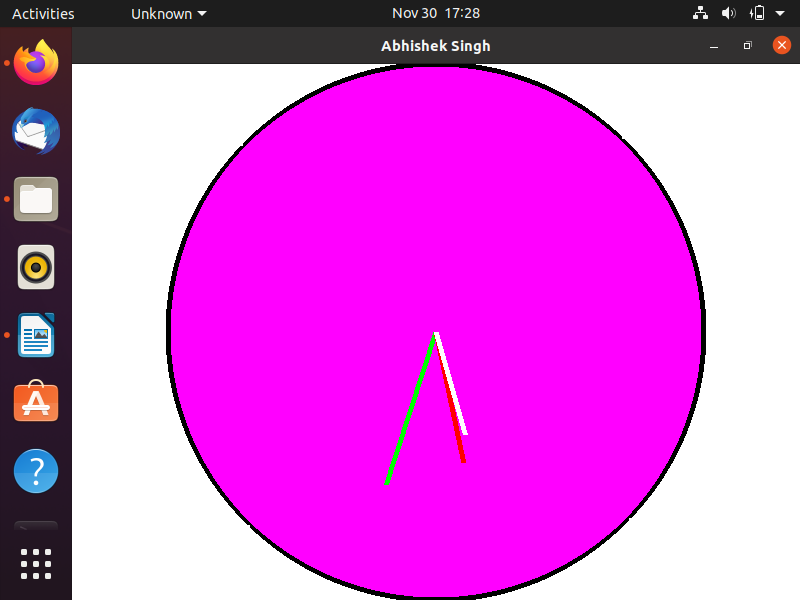
**initialize();**

**glutMainLoop();**

**return 0;**

**}**

**Output:**



**17.Cohen Sutherland Clipping**

**#include <GL/glut.h>**

**#define SCREEN\_WIDTH 640**

**#define SCREEN\_HEIGHT 480**

**typedef struct {**

**GLfloat x, y;**

**} Point;**

**const GLint WIN\_LEFT\_BIT = 0x01;**

**const GLint WIN\_RIGHT\_BIT = 0x02;**

**const GLint WIN\_BOTTOM\_BIT = 0x04;**

**const GLint WIN\_TOP\_BIT = 0x08;**

**void init\_graph(int argc, char \*\*argv) {**

**glutInit(&argc, argv);**

**glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);**

**glutInitWindowSize(SCREEN\_WIDTH, SCREEN\_HEIGHT);**

**glutCreateWindow(argv[0]);**

**glClearColor(1.0, 1.0, 1.0, 0.0);**

**glPointSize(1.0f);**

**glMatrixMode(GL\_PROJECTION);**

**glLoadIdentity();**

**gluOrtho2D(0, SCREEN\_WIDTH, 0, SCREEN\_HEIGHT);**

**}**

**void close\_graph() {**

**glutMainLoop();**

**}**

**void swap\_points(Point \*p1, Point \*p2) {**

**Point t = \*p1;**

**\*p1 = \*p2;**

**\*p2 = t;**

**}**

**void swap\_codes(GLint \*x, GLint \*y) {**

**GLint t = \*x;**

**\*x = \*y;**

**\*y = t;**

**}**

**GLint inside(GLint code) {**

**return !code;**

**}**

**GLint accept(GLint code1, GLint code2) {**

**return !(code1 | code2);**

**}**

**GLint reject(GLint code1, GLint code2) {**

**return code1 & code2;**

**}**

**GLint encode(Point p1, Point win\_min, Point win\_max) {**

**GLint code = 0x00;**

**if (p1.x < win\_min.x) code |= WIN\_LEFT\_BIT;**

**if (p1.x > win\_max.x) code |= WIN\_RIGHT\_BIT;**

**if (p1.y < win\_min.y) code |= WIN\_BOTTOM\_BIT;**

**if (p1.y > win\_max.y) code |= WIN\_TOP\_BIT;**

**return code;**

**}**

**GLint round(GLfloat a) {**

**return (GLint) (a + 0.5f);**

**}**

**void line\_clip(Point p1, Point p2, Point win\_min, Point win\_max) {**

**GLint code1, code2;**

**GLint done = 0, plot\_line = 0;**

**GLfloat m = 0;**

**if (p1.x != p2.x) {**

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

**}**

**while (!done) {**

**code1 = encode(p1, win\_min, win\_max);**

**code2 = encode(p2, win\_min, win\_max);**

**if (accept(code1, code2)) {**

**done = 1;**

**plot\_line = 1;**

**} else if (reject(code1, code2)) {**

**done = 1;**

**} else {**

**if (inside(code1)) {**

**swap\_points(&p1, &p2);**

**swap\_codes(&code1, &code2);**

**}**

**if (code1 & WIN\_LEFT\_BIT) {**

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

**p1.x = win\_min.x;**

**} else if (code1 & WIN\_RIGHT\_BIT) {**

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

**p1.x = win\_max.x;**

**} else if (code1 & WIN\_BOTTOM\_BIT) {**

**if (p1.x != p2.x)**

**p1.x += (win\_min.y - p1.y) / m;**

**p1.y = win\_min.y;**

**} else if (code1 & WIN\_TOP\_BIT) {**

**if (p1.x != p2.x)**

**p1.x += (win\_max.y - p1.y) / m;**

**p1.y = win\_max.y;**

**}**

**}**

**}**

**if (plot\_line) {**

**glColor3f(1, 0, 0);**

**glLineWidth(2);**

**glBegin(GL\_LINES);**

**glVertex2i(round(p1.x), round(p1.y));**

**glVertex2i(round(p2.x), round(p2.y));**

**glEnd();**

**glFlush();**

**}**

**}**

**void draw\_window(Point win\_min, Point win\_max) {**

**glColor3f(1, 0, 0);**

**glBegin(GL\_LINE\_LOOP);**

**glVertex2i(round(win\_min.x), round(win\_min.y));**

**glVertex2i(round(win\_min.x), round(win\_max.y));**

**glVertex2i(round(win\_max.x), round(win\_max.y));**

**glVertex2i(round(win\_max.x), round(win\_min.y));**

**glEnd();**

**glFlush();**

**}**

**void init\_clip() {**

**glClear(GL\_COLOR\_BUFFER\_BIT);**

**Point win\_min = {50, 50};**

**Point win\_max = {450, 450};**

**draw\_window(win\_min, win\_max);**

**Point p1 = {25, 500};**

**Point p2 = {500, 300};**

**glColor3f(0.5, 0.5, 0.5);**

**glBegin(GL\_LINES);**

**glVertex2i(round(p1.x), round(p1.y));**

**glVertex2i(round(p2.x), round(p2.y));**

**glEnd();**

**line\_clip(p1, p2, win\_min, win\_max);**

**}**

**int main(int argc, char \*\*argv) {**

**init\_graph(argc, argv);**

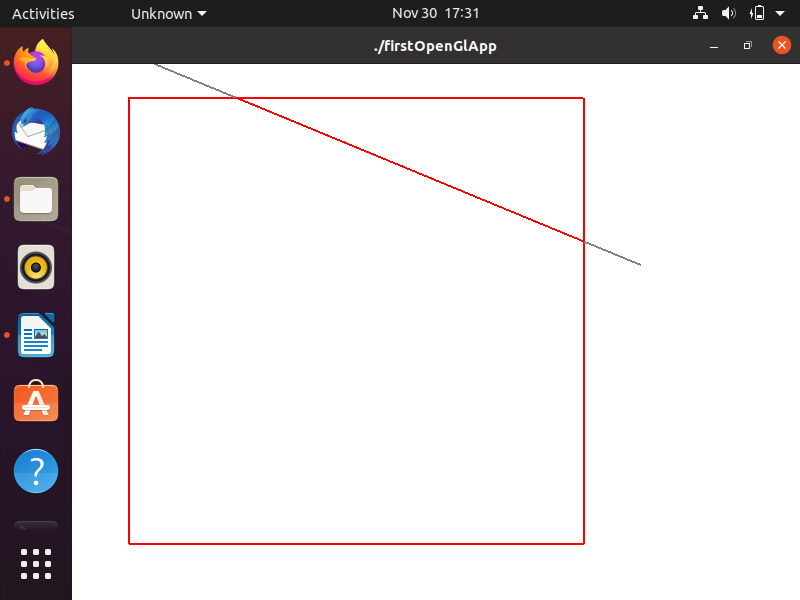
**glutDisplayFunc(init\_clip);**

**close\_graph();**

**return 0;**

**}**

**Output:**

****

**18. 3D Transformation**

**#include<iostream>**

**#include<stdlib.h>**

**#include<GL/glut.h>**

**using namespace std;**

**#define KEY\_ESC 27**

**#define KEY\_SPC 32**

**#define S 83**

**#define R 82**

**#define T 84**

**float chx=1.0,chy=1.0,chz=1.0;**

**float tx=0.0,ty=0.5,tz=0.0;**

**float \_angle = 30.0f;**

**float rx=0.0,ry=1.0,rz=1.0;**

**void onScale(float,float,float);**

**void cube(void)**

**{**

**glBegin(GL\_POLYGON);**

**glColor3f(0.0, 0.0, 1.0);**

**glVertex3f(0.5, -0.5, 0.5 );**

**glColor3f(1.0, 0.0, 0.0 );**

**glVertex3f(0.5, 0.5, 0.5 );**

**glColor3f(0.6, 0.6, 0.6);**

**glVertex3f(-0.5, 0.5, 0.5 );**

**glColor3f(0.2, 0.2, 0.2 );**

**glVertex3f(-0.5, -0.5, 0.5 );**

**glEnd();**

**// Purple side - RIGHT**

**glBegin(GL\_POLYGON);**

**glColor3f( 1.0, 0.0, 1.0 );**

**glVertex3f( 0.5, -0.5, -0.5 );**

**glColor3f( 0.5, 0.0, 0.5);**

**glVertex3f( 0.5, 0.5, -0.5 );**

**glColor3f( 0.7, 0.0, 0.7 );**

**glVertex3f( 0.5, 0.5, 0.5 );**

**glColor3f( 0.3, 0.0, 0.3 );**

**glVertex3f( 0.5, -0.5, 0.5 );**

**glEnd();**

**// Green side - LEFT**

**glBegin(GL\_POLYGON);**

**glColor3f( 0.0, 1.0, 0.0 );**

**glVertex3f( -0.5, -0.5, 0.5 );**

**glColor3f( 0.0, 0.3, 0.0 );**

**glVertex3f( -0.5, 0.5, 0.5 );**

**glColor3f( 0.0, 0.6, 0.0 );**

**glVertex3f( -0.5, 0.5, -0.5 );**

**glColor3f( 0.0, 0.7, 0.0 );**

**glVertex3f( -0.5, -0.5, -0.5 );**

**glEnd();**

**// Blue side - TOP**

**glBegin(GL\_POLYGON);**

**glColor3f( 0.0, 0.0, 1.0 );**

**glVertex3f( 0.5, 0.5, 0.5 );**

**glColor3f( 0.0, 0.0, 0.7 );**

**glVertex3f( 0.5, 0.5, -0.5 );**

**glColor3f( 0.0, 0.0, 0.5 );**

**glVertex3f( -0.5, 0.5, -0.5 );**

**glColor3f( 0.0, 0.0, 0.3 );**

**glVertex3f( -0.5, 0.5, 0.5 );**

**glEnd();**

**// Red side - BOTTOM**

**glBegin(GL\_POLYGON);**

**glColor3f( 1.0, 0.0, 0.0 );**

**glVertex3f( 0.5, -0.5, -0.5 );**

**glColor3f( 0.7, 0.0, 0.0 );**

**glVertex3f( 0.5, -0.5, 0.5 );**

**glColor3f( 0.5, 0.0, 0.0 );**

**glVertex3f( -0.5, -0.5, 0.5 );**

**glColor3f( 0.3, 0.0, 0.0 );**

**glVertex3f( -0.5, -0.5, -0.5 );**

**glEnd();**

**glFlush();**

**glutSwapBuffers();**

**}**

**void initRenderin()**

**{**

**glEnable(GL\_DEPTH\_TEST);**

**}**

**void handleResize(int w,int h)**

**{**

**glViewport(0,0,w,h);**

**glMatrixMode(GL\_PROJECTION);**

**glLoadIdentity();**

**gluPerspective(45.0,(double)w / (double)h,1.0,200.0);**

**}**

**float \_cameraAngle = 0.0f;**

**void drawScene(void)**

**{**

**glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT);**

**glMatrixMode(GL\_MODELVIEW);**

**glLoadIdentity();**

**glRotatef(-\_cameraAngle,0.0f,1.0f,0.0f);**

**glTranslatef(0.0f,0.0f,-5.0f);**

**glPushMatrix();**

**glTranslatef(tx,ty,tz);**

**glRotatef(\_angle,rx,ry,rz);**

**glScalef(chx,chy,chz);**

**cube();**

**/\*glBegin(GL\_QUADS);**

**glColor3f(1.0f,0.0f,0.0f);**

**glVertex3f(-0.5,0.5,0);**

**glColor3f(0.5f,0.0f,0.0f);**

**glVertex3f(-0.5,-0.5,0);**

**glColor3f(0.8f,0.0f,0.0f);**

**glVertex3f(0.5,-0.5,0);**

**glColor3f(0.2f,0.0f,0.0f);**

**glVertex3f(0.5,0.5,0);**

**glEnd();\*/**

**glPopMatrix();**

**glutSwapBuffers();**

**}**

**void update(int value)**

**{**

**\_angle+=2.0f;**

**if(\_angle>360){**

**\_angle-=360;**

**}**

**glutPostRedisplay();**

**glutTimerFunc(25,update,0);**

**}**

**GLvoid window\_key(unsigned char key, int x, int y)**

**{**

**switch (key) {**

**case KEY\_ESC:**

**exit(1);**

**break;**

**case S:**

**printf("enter scaling factors :");**

**scanf("%f%f%f",&chx,&chy,&chz);**

**break;**

**case T:**

**printf("enter translation factors :");**

**scanf("%f%f%f",&tx,&ty,&tz);**

**break;**

**case R:**

**printf("enter rotation angle :");**

**scanf("%f",&\_angle);**

**printf("enter rotation factor :");**

**scanf("%f%f%f",&rx,&ry,&rz);**

**break;**

**default:**

**printf ("Pressing %d doesn't do anything.\n", key);**

**break;**

**}**

**}**

**int main(int argc,char\*\* argv)**

**{**

**glutInit(&argc,argv);**

**glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_RGB | GLUT\_DEPTH);**

**glutInitWindowSize(400,400);**

**glutCreateWindow("3D Transfomration");**

**initRenderin();**

**glutDisplayFunc(drawScene);**

**glutReshapeFunc(handleResize);**

**glutKeyboardFunc(&window\_key);**

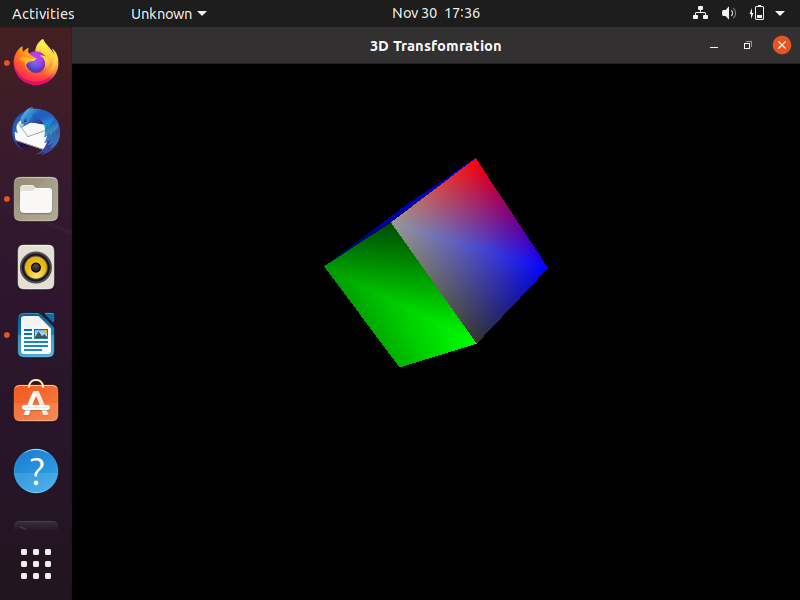
**glutTimerFunc(25,update,0);**

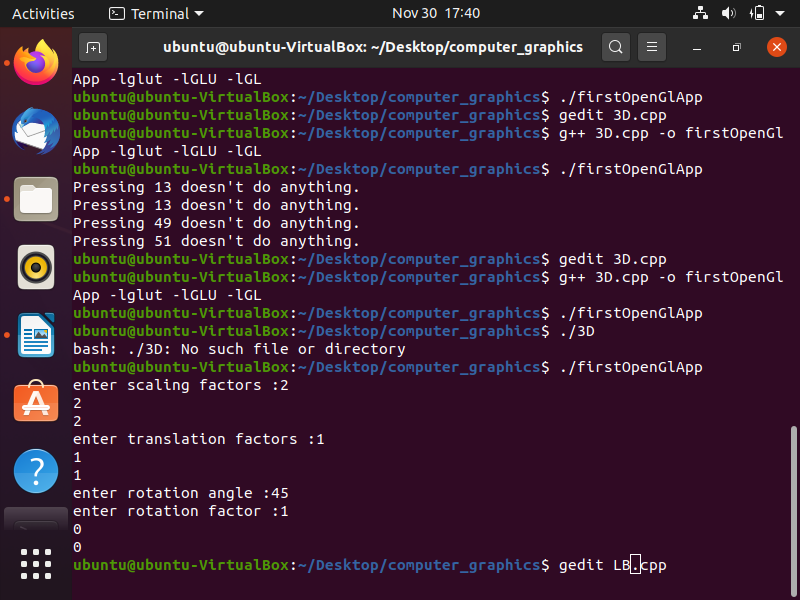
**glutMainLoop();**

**return 0;**

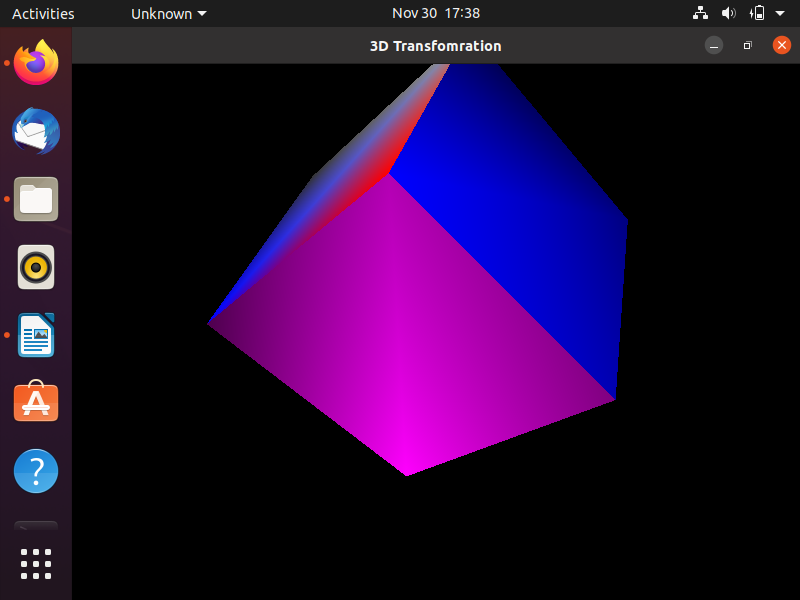
**}**

**Output:**

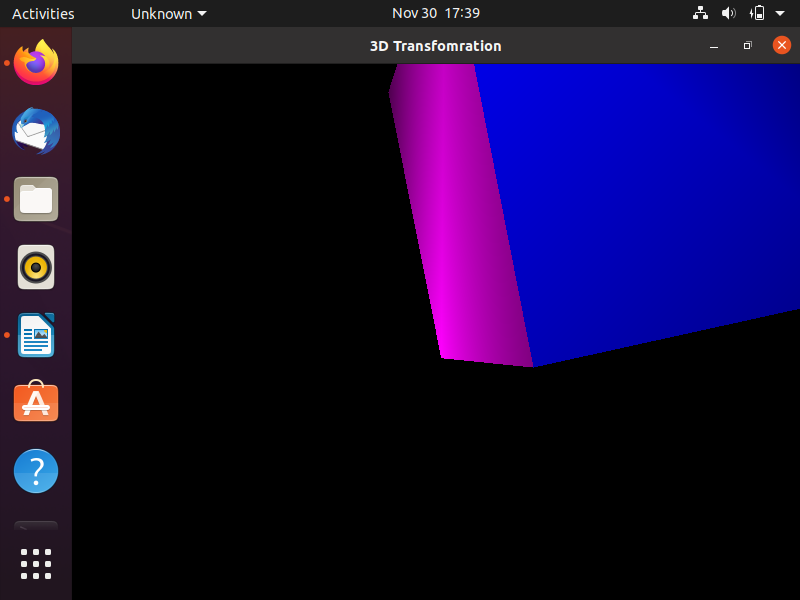




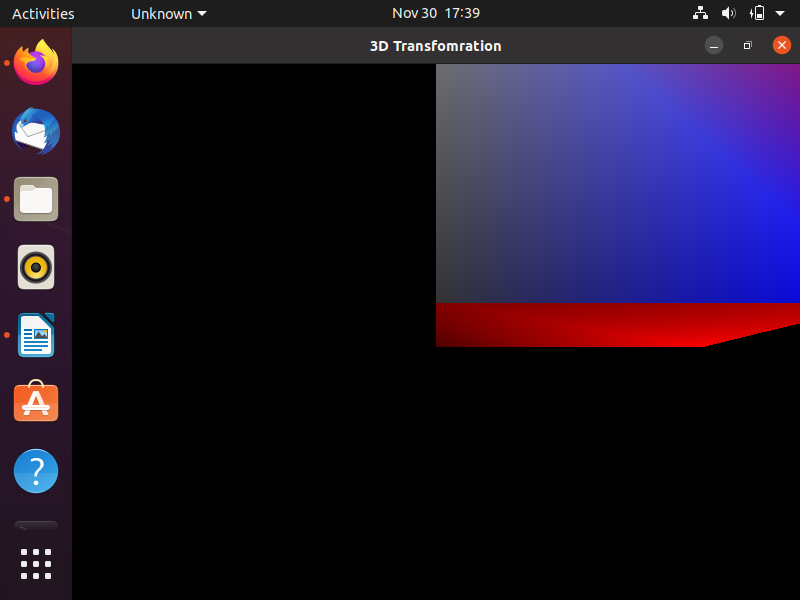
**Scaling:**



**Translation:**



**Rotation:**



**19.Object Traversing With Arrow Keys**

**#include <stdio.h>**

**#include <GL/glut.h>**

**GLfloat rotation = 90.0;**

**float posX = 0, posY = 0, posZ = 0;**

**void reshape(int width, int heigth){**

**/\* window ro reshape when made it bigger or smaller\*/**

**glMatrixMode(GL\_PROJECTION);**

**glLoadIdentity();**

**//clip the windows so its shortest side is 2.0**

**if (width < heigth) {**

**glOrtho(-2.0, 2.0, -2.0 \* (GLfloat)heigth / (GLfloat)width, 2.0 \* (GLfloat)heigth / (GLfloat)width, 2.0, 2.0);**

**}**

**else{**

**glOrtho(-2.0, 2.0, -2.0 \* (GLfloat)width / (GLfloat)heigth, 2.0 \* (GLfloat)width / (GLfloat)heigth,2.0 , 2.0);**

**}**

**// set viewport to use the entire new window**

**glViewport(0, 0, width, heigth);**

**}**

**void rect(){**

**glBegin(GL\_POLYGON);**

**glColor3f(1.0, 0.5, 1.0);**

**glVertex2f(-0.1, 0.0);**

**glVertex2f(0.0, 0.2);**

**glVertex2f(0.1, 0.0);**

**glVertex2f(0.0, -0.2);**

**glEnd();**

**}**

**void display(){**

**//Clear Window**

**glClear(GL\_COLOR\_BUFFER\_BIT);**

**glMatrixMode(GL\_MODELVIEW);**

**glLoadIdentity();**

**glPushMatrix();**

**glTranslatef(posX,posY,posZ);**

**rect();**

**glPopMatrix();**

**glFlush();**

**}**

**void init(){**

**// set clear color to black**

**glClearColor(0.0, 0.0, 0.0, 0.0);**

**// set fill color to white**

**glColor3f(1.0, 1.0, 1.0);**

**//set up standard orthogonal view with clipping**

**//box as cube of side 2 centered at origin**

**//This is the default view and these statements could be removed**

**glMatrixMode(GL\_PROJECTION);**

**glLoadIdentity();**

**gluOrtho2D(-1.0, 1.0, -1.0, 1.0);**

**}**

**float move\_unit = 0.1f;**

**void keyboardown(int key, int x, int y)**

**{**

**switch (key){**

**case GLUT\_KEY\_RIGHT:**

**posX+=move\_unit;;**

**break;**

**case GLUT\_KEY\_LEFT:**

**posX-=move\_unit;;**

**break;**

**case GLUT\_KEY\_UP:**

**posY+=move\_unit;;**

**break;**

**case GLUT\_KEY\_DOWN:**

**posY-=move\_unit;;**

**break;**

**default:**

**break;**

**}**

**glutPostRedisplay();**

**}**

**int main(int argc, char\*\* argv){**

**//initialize mode and open a windows in upper left corner of screen**

**//Windows tittle is name of program**

**glutInit(&argc, argv);**

**glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);**

**glutInitWindowSize(500,500);**

**glutInitWindowPosition(0, 0);**

**glutCreateWindow("Abhishek Sharma");**

**glutDisplayFunc(display);**

**init();**

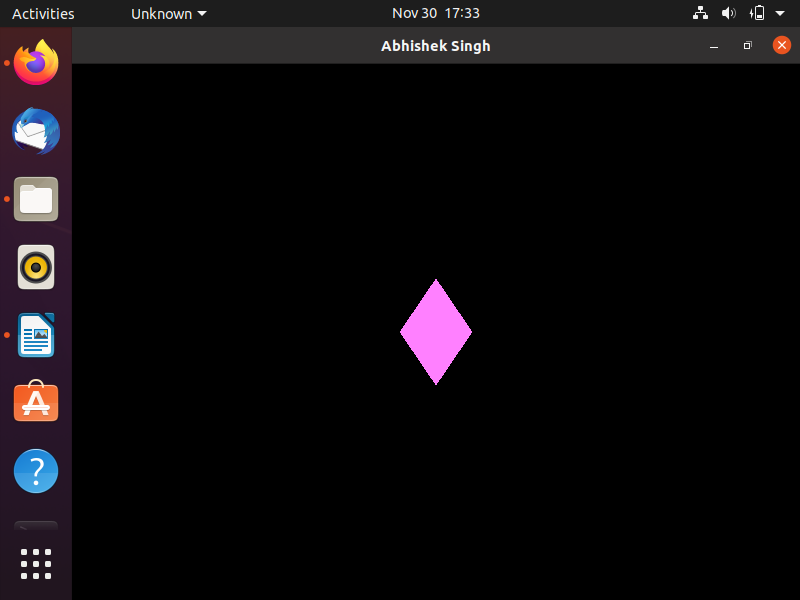
**glutSpecialFunc(keyboardown);**

**glutMainLoop();**

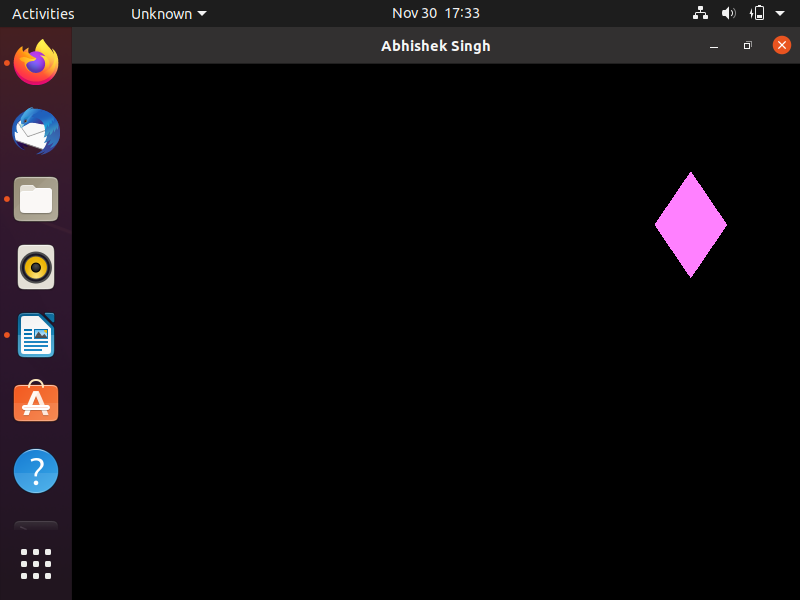
**}**

**Output:**

**Then**

****

**Now:-**

****

**20.Liang Barsky**

**#include <GL/glut.h>**

**double y\_max = 150, y\_min = 100, x\_max = 125, x\_min = 75; // Old viewport**

**double ny\_max = 300, ny\_min = 200, nx\_max = 300, nx\_min = 200; // New clipped ViewPort**

**double t1 = 0.0, t2 = 1.0; // Intial and final time**

**double x1 = 90, y1 = 175; // Point 1**

**double x2 = 150, y2 = 101; // Point 2**

**void myDisplay();**

**void draw\_lineAndPort(double x1, double y1, double x2, double y2, double y\_max, double y\_min, double x\_max, double x\_min);**

**void liangBarsky(double x1, double y1, double x2, double y2);**

**bool cliptest(double p, double q);**

**void myInit()**

**{**

**glLoadIdentity();**

**glMatrixMode(GL\_PROJECTION);**

**gluOrtho2D(0, 500, 0, 500);**

**glMatrixMode(GL\_MODELVIEW);**

**}**

**int main(int argc, char \*\*argv)**

**{**

**glutInit(&argc, argv);**

**glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);**

**glutInitWindowPosition(0, 0);**

**glutInitWindowSize(500, 500);**

**glutCreateWindow("Abhishek Sharma");**

**glutDisplayFunc(myDisplay);**

**myInit();**

**glutMainLoop();**

**return 0;**

**}**

**void myDisplay()**

**{**

**glClear(GL\_COLOR\_BUFFER\_BIT);**

**draw\_lineAndPort(x1, y1, x2, y2, y\_max, y\_min, x\_max, x\_min);**

**liangBarsky(x1, y1, x2, y2);**

**glFlush();**

**}**

**void draw\_lineAndPort(double x1, double y1, double x2, double y2, double y\_max, double y\_min, double x\_max, double x\_min)**

**{**

**glColor3d(1, 1, 1);**

**glBegin(GL\_LINE\_LOOP);**

**glVertex2d(x\_min, y\_min);**

**glVertex2d(x\_max, y\_min);**

**glVertex2d(x\_max, y\_max);**

**glVertex2d(x\_min, y\_max);**

**glEnd();**

**glColor3d(1, 0, 1);**

**glBegin(GL\_LINES);**

**glVertex2d(x1, y1);**

**glVertex2d(x2, y2);**

**glEnd();**

**}**

**bool cliptest(double p, double q)**

**{**

**double t = q / p;**

**if (p == 0 && q < 0) // Line is parallel to viewport and outside**

**{**

**return false;**

**}**

**else if (p < 0)**

**{**

**if (t > t1)**

**t1 = t;**

**if (t > t2)**

**return false;**

**}**

**else if (p > 0)**

**{**

**if (t < t2)**

**t2 = t;**

**if (t < t1)**

**return false;**

**}**

**return true;**

**}**

**void liangBarsky(double x1, double y1, double x2, double y2)**

**{**

**double dx = x2 - x1;**

**double dy = y2 - y1;**

**/\***

**-t \* dx < x1 - x\_min ... [1]**

**t \* dx < x\_max - x1 ... [2]**

**-t \* dy < y1 - y\_min ... [3]**

**t \* dy < y\_max - y1 ... [4]**

**\*/**

**if (cliptest(-dx, x1 - x\_min) && cliptest(dx, x\_max - x1) && cliptest(-dy, y1 - y\_min) && cliptest(dy, y\_max - y1))**

**{**

**if (t2 < 1)**

**{**

**x2 = x1 + t2 \* dx;**

**y2 = y1 + t2 \* dy;**

**}**

**if (t1 > 0)**

**{**

**x1 = x1 + t1 \* dx;**

**y1 = y1 + t1 \* dy;**

**}**

**// Scaling to new View port**

**double scale\_x = (nx\_max - nx\_min) / (x\_max - x\_min);**

**double scale\_y = (ny\_max - ny\_min) / (y\_max - y\_min);**

**// New coordinates of the points**

**// Point 1**

**double nx1 = nx\_min + (x1 - x\_min) \* scale\_x;**

**double ny1 = ny\_min + (y1 - y\_min) \* scale\_y;**

**// Point 2**

**double nx2 = nx\_min + (x2 - x\_min) \* scale\_x;**

**double ny2 = ny\_min + (y2 - y\_min) \* scale\_y;**

**// Plotting new Viewport and clipped line**

**draw\_lineAndPort(nx1, ny1, nx2, ny2, ny\_max, ny\_min, nx\_max, nx\_min);**

**}**

**}**

**Output:-**

