**CGG ASSIGNMENT 4**

Name: Kedar Kale

Roll No: 129

Block B1

Topic: Cohen-Sutherland Line Clipping Algorithm

Code:

#include<stdio.h>

#include<GL/glut.h>

#include<stdbool.h>

int TOP = 8;

int BOTTOM = 4;

int LEFT = 1;

int RIGHT = 2;

int INSIDE = 0;

int x\_min = 100;

int x\_max = 400;

int y\_min = 100;

int y\_max = 400;

int pt\_x1, pt\_x2, pt\_y1, pt\_y2;

void init2D(float r, float g, float b)

{

glClear(GL\_COLOR\_BUFFER\_BIT);

glClearColor(r,g,b,0.0);

glMatrixMode (GL\_PROJECTION);

gluOrtho2D (0.0, 500.0, 500.0, 0.0);

}

int cal\_code(int x, int y)

{

int code = INSIDE;

if(x < x\_min)

{

code = code | LEFT;

}

else if(x > x\_max)

{

code = code | RIGHT;

}

else if(y < y\_min)

{

code = code | BOTTOM;

}

else if(y > y\_max)

{

code = code | TOP;

}

return code;

}

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

{

int code1 = cal\_code(x1,y1);

int code2 = cal\_code(x2,y2);

bool accept = false;

while(true)

{

if(code1 == 0 && code2 == 0)

{

accept = true;

break;

}

else if(code1 & code2)

{

break;

}

else

{

int code\_out;

int x, y;

if(code1 != 0)

{

code\_out = code1;

}

else

{

code\_out = code2;

}

if(code\_out & BOTTOM)

{

x = x1 + (x2 - x1) \* (y\_min - y1) / (y2 - y1);

y = y\_min;

}

else if(code\_out & TOP)

{

x = x1 + (x2 - x1) \* (y\_max - y1) / (y2 - y1);

y = y\_max;

}

else if(code\_out & RIGHT)

{

y = y1 + (y2 - y1) \* (x\_max - x1) / (x2 - x1);

x = x\_max;

}

else if(code\_out & LEFT)

{

y = y1 + (y2 - y1) \* (x\_min - x1) / (x2 - x1);

x = x\_min;

}

if(code\_out == code1)

{

x1 = x;

y1 = y;

code1 = cal\_code(x1,y1);

}

else

{

x2 = x;

y2 = y;

code2 = cal\_code(x2,y2);

}

}

}

if(accept)

{

glColor3f(0.0,1.0,0.0);

glBegin(GL\_LINES);

glVertex2i(x1,y1);

glVertex2i(x2,y2);

glEnd();

glFlush();

}

}

void display()

{

clipping(pt\_x1, pt\_y1, pt\_x2, pt\_y2);

glBegin(GL\_LINES);

glColor3f(0.0,1.0,0.0);

glVertex2i(x\_min,y\_min);

glVertex2i(x\_min,y\_max);

glVertex2i(x\_max,y\_min);

glVertex2i(x\_max,y\_max);

glVertex2i(x\_min, y\_max);

glVertex2i(x\_max,y\_max);

glVertex2i(x\_min,y\_min);

glVertex2i(x\_max,y\_min);

glEnd();

glFlush();

}

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

{

glutInit(&argc, argv);

scanf("%d", &pt\_x1);

scanf("%d", &pt\_y1);

scanf("%d", &pt\_x2);

scanf("%d", &pt\_y2);

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

glutCreateWindow("Clipping\_Box");

glutInitWindowSize(1000,1000);

glutInitWindowPosition(0,0);

init2D(0.0,0.0,0.0);

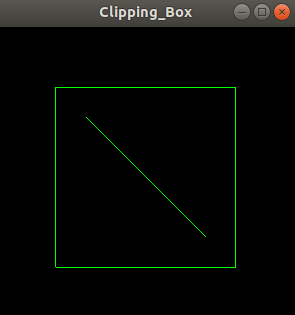
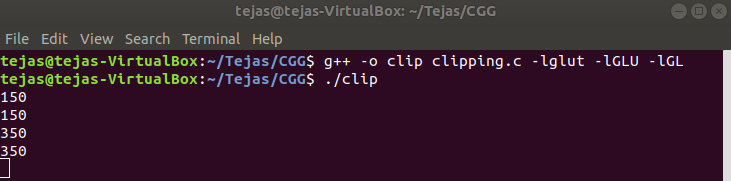
glutDisplayFunc(display);

glutMainLoop();

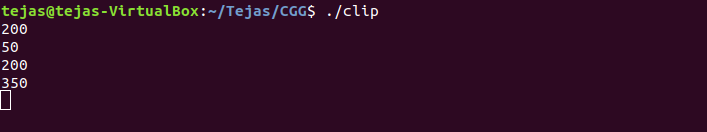
}

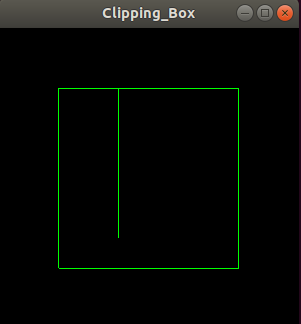
Output:

1) Completely inside the box:



2) Paritally inside the box:





3) Completely outside the box:

