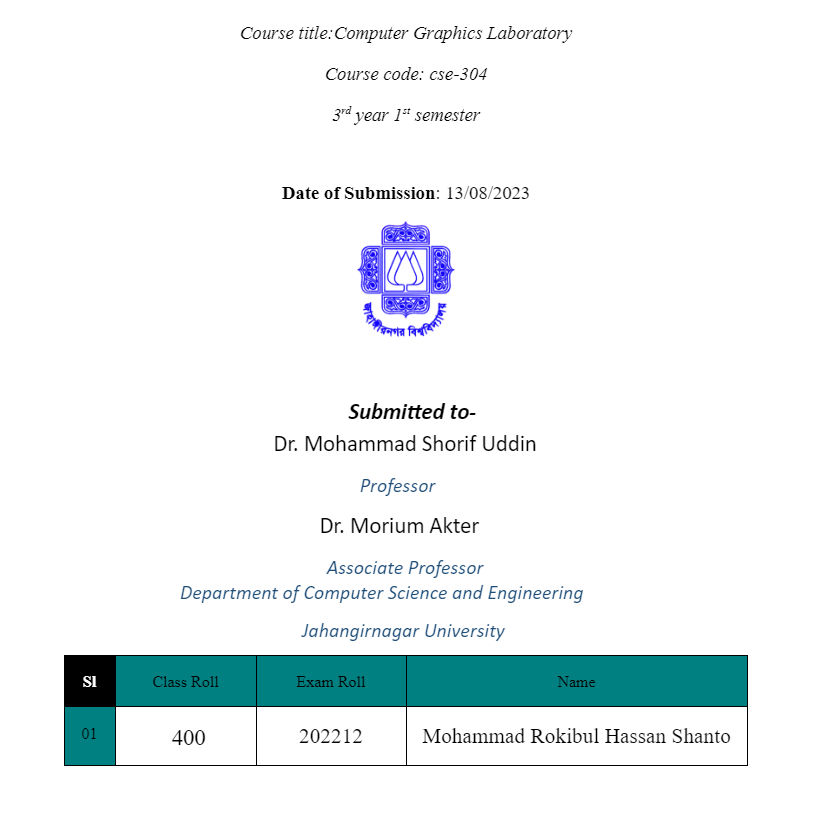
****

**Liang barsky algorithm:**

**Code1:**

#include <iostream>

#include <graphics.h>

void liangBarsky(double x1, double y1, double x2, double y2, double xmin, double ymin, double xmax, double ymax) {

double dx = x2 - x1, dy = y2 - y1;

double p[4] = {-dx, dx, -dy, dy};

double q[4] = {x1 - xmin, xmax - x1, y1 - ymin, ymax - y1};

double t\_in = 0.0, t\_out = 1.0;

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

if (p[i] == 0 && q[i] < 0) {

std::cout << "Line is outside of the clipping window." << std::endl;

return;

} else if (p[i] != 0) {

double t = q[i] / p[i];

if (p[i] < 0) {

t\_in = std::max(t\_in, t);

} else if (p[i] > 0) {

t\_out = std::min(t\_out, t);

}

}

}

if (t\_in > t\_out) {

std::cout << "Line is outside of the clipping window." << std::endl;

} else {

double new\_x1 = x1 + t\_in \* dx;

double new\_y1 = y1 + t\_in \* dy;

double new\_x2 = x1 + t\_out \* dx;

double new\_y2 = y1 + t\_out \* dy;

line(new\_x1, new\_y1, new\_x2, new\_y2);

}

}

int main() {

int gd = DETECT, gm;

initgraph(&gd, &gm, "");

double x1 = 50, y1 = 50, x2 = 200, y2 = 200;

double xmin = 100, ymin = 100, xmax = 300, ymax = 300;

rectangle(xmin, ymin, xmax, ymax);

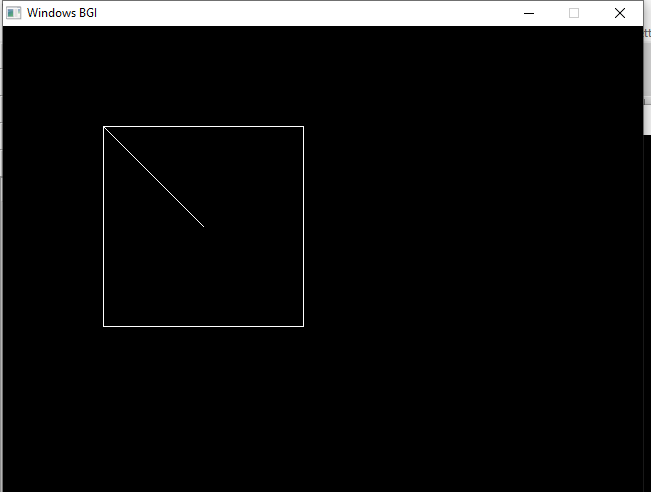
liangBarsky(x1, y1, x2, y2, xmin, ymin, xmax, ymax);

delay(5000); // Pause for 5 seconds

closegraph();

return 0;

}



**Code 2:**

#include <iostream>

#include <cmath>

#include <graphics.h>

using namespace std;

const int LEFT = 1, RIGHT = 2, BOTTOM = 4, TOP = 8;

int x\_min, y\_min, x\_max, y\_max;

int computeCode(int x, int y) {

int code = 0;

if (x < x\_min)

code |= LEFT;

if (x > x\_max)

code |= RIGHT;

if (y < y\_min)

code |= BOTTOM;

if (y > y\_max)

code |= TOP;

return code;

}

void liangBarsky(int x1, int y1, int x2, int y2) {

int code1 = computeCode(x1, y1);

int code2 = computeCode(x2, y2);

bool accept = false;

while (true) {

if (!(code1 | code2)) {

accept = true;

break;

} else if (code1 & code2) {

break;

} else {

int codeOut = code1 ? code1 : code2;

int x, y;

if (codeOut & TOP) {

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

y = y\_max;

} else if (codeOut & BOTTOM) {

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

y = y\_min;

} else if (codeOut & RIGHT) {

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

x = x\_max;

} else if (codeOut & LEFT) {

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

x = x\_min;

}

if (codeOut == code1) {

x1 = x;

y1 = y;

code1 = computeCode(x1, y1);

} else {

x2 = x;

y2 = y;

code2 = computeCode(x2, y2);

}

}

}

if (accept) {

line(x1, y1, x2, y2);

}

}

int main() {

int gd = DETECT, gm;

initgraph(&gd, &gm, "");

cout << "Enter the coordinates of the clipping window (x\_min y\_min x\_max y\_max): ";

cin >> x\_min >> y\_min >> x\_max >> y\_max;

rectangle(x\_min, y\_min, x\_max, y\_max);

int x1, y1, x2, y2;

cout << "Enter the endpoints of the line (x1 y1 x2 y2): ";

cin >> x1 >> y1 >> x2 >> y2;

line(x1, y1, x2, y2);

delay(10000);

closegraph();

return 0;

}

**Code3:**

#include <iostream>

#include <cmath>

#include <graphics.h>

using namespace std;

const int LEFT = 1, RIGHT = 2, BOTTOM = 4, TOP = 8;

int x\_min, y\_min, x\_max, y\_max;

int computeCode(int x, int y) {

int code = 0;

if (x < x\_min)

code |= LEFT;

if (x > x\_max)

code |= RIGHT;

if (y < y\_min)

code |= BOTTOM;

if (y > y\_max)

code |= TOP;

return code;

}

void liangBarsky(int x1, int y1, int x2, int y2) {

int code1 = computeCode(x1, y1);

int code2 = computeCode(x2, y2);

bool accept = false;

while (true) {

if (!(code1 | code2)) {

accept = true;

break;

} else if (code1 & code2) {

break;

} else {

int codeOut = code1 ? code1 : code2;

int x, y;

if (codeOut & TOP) {

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

y = y\_max;

} else if (codeOut & BOTTOM) {

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

y = y\_min;

} else if (codeOut & RIGHT) {

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

x = x\_max;

} else if (codeOut & LEFT) {

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

x = x\_min;

}

if (codeOut == code1) {

x1 = x;

y1 = y;

code1 = computeCode(x1, y1);

} else {

x2 = x;

y2 = y;

code2 = computeCode(x2, y2);

}

}

}

if (accept) {

line(x1, y1, x2, y2);

}

}

int main() {

int gd = DETECT, gm;

initgraph(&gd, &gm, "");

cout << "Enter the coordinates of the clipping window (x\_min y\_min x\_max y\_max): ";

cin >> x\_min >> y\_min >> x\_max >> y\_max;

rectangle(x\_min, y\_min, x\_max, y\_max);

int numLines;

cout << "Enter the number of lines to clip: ";

cin >> numLines;

for (int i = 0; i < numLines; i++) {

int x1, y1, x2, y2;

cout << "Enter endpoints of line " << (i + 1) << " (x1 y1 x2 y2): ";

cin >> x1 >> y1 >> x2 >> y2;

liangBarsky(x1, y1, x2, y2);

}

delay(10000);

closegraph();

return 0;

}

Output:

