

CODE:

```
#include <iostream>

#include <graphics.h>

#include <math.h>

#include <cstdlib>

using namespace std;

static int LEFT = 1, RIGHT = 2, BOTTOM = 4, TOP = 8, xmax, ymax, xmin, ymin;

int getcode(int x, int y) {

    int code = 0;

    if (y > ymax) code |= TOP;

    if (y < ymin) code |= BOTTOM;

    if (x < xmin) code |= LEFT;

    if (x > xmax) code |= RIGHT;

    return code;

}

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

    char coords[100];

    sprintf(coords, "Line Endpoints: (%d, %d) to (%d, %d)", x1, y1, x2, y2);

    outtextxy(10, 10, coords);

    delay(3000); // Display for 3 seconds

}

int main() {

    int gd = DETECT, gm;

    int x1, y1, x2, y2;

    cout << "Enter top left and bottom right coordinates: ";
```

```

cin >> xmin >> ymin >> xmax >> ymax;

cout << "Enter endpoints of line: ";

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

initgraph(&gd, &gm, NULL);

// Draw the clipping rectangle
rectangle(xmin, ymin, xmax, ymax);

// Show input coordinates

setcolor(WHITE); // Set color for text

showCoordinates(x1, y1, x2, y2);


// Draw the original line
line(x1, y1, x2, y2);

// Delay to show the original line
delay(2000); // Display for 2 seconds

// Clipping

int outcode1 = getcode(x1, y1);

int outcode2 = getcode(x2, y2);

bool accept = false;

while (true) {

    if ((outcode1 == 0) && (outcode2 == 0)) {

        accept = true;

        break;

    } else if ((outcode1 & outcode2) != 0) {

        break;

    } else {

```

```

int x, y;

int temp;

if (outcode1 != 0) {

    temp = outcode1;

} else {

    temp = outcode2;

}

if (temp & TOP) {

     $x = x1 + (x2 - x1) * (ymax - y1) / (y2 - y1);$ 

    y = ymax;

} else if (temp & BOTTOM) {

     $x = x1 + (x2 - x1) * (ymin - y1) / (y2 - y1);$ 

    y = ymin;

} else if (temp & LEFT) {

     $y = y1 + (y2 - y1) * (xmin - x1) / (x2 - x1);$ 

    x = xmin;

} else if (temp & RIGHT) {

     $y = y1 + (y2 - y1) * (xmax - x1) / (x2 - x1);$ 

    x = xmax;

}

if (temp == outcode1) {

    x1 = x;

    y1 = y;

    outcode1 = getcode(x1, y1);

} else {

```

```

        x2 = x;

        y2 = y;

        outcode2 = getcode(x2, y2);

    }

}

cleardevice(); // Clear the screen

rectangle(xmin, ymin, xmax, ymax); // Redraw the clipping rectangle

setcolor(BROWN); // Set the color for the clipped line

if (accept) {

    line(x1, y1, x2, y2); // Draw the clipped line

    // Show the new coordinates after clipping

    char clippedCoords[100];

    sprintf(clippedCoords, "Clipped Line Endpoints: (%d, %d) to (%d, %d)", x1, y1, x2, y2);

    outtextxy(10, 10, clippedCoords);

} else {

    outtextxy(10, 10, "Line is completely outside the clipping rectangle."); // Provide a message

}

delay(3000); // Wait for 3 seconds before closing

getch(); // Wait for a key press

closegraph(); // Close the graphics window

return 0;

}

```

USER INPUT:

```

Enter top left and bottom right coordinates: 200 200 500 500
Enter endpoints of line: 150 150 350 350

```

OUTPUT:

