Harsh Kashiwal
2003085
C21

Page No.
Date

Experiment 7

Aim: Implement Liang Barsky line clipping.

Theory:

Algorithm:

Step 1: Take $x_{min}, x_{max}, y_{min}, y_{max}, x_1, y_1, x_2, y_2$ as input from the user and set $t_1 = 0$ $t_2 = 1$

Step 2: Calculate $\Delta x = x_2 - x_1$, $\Delta y = y_2 - y_1$

$p_1 = -\Delta x$, $q_1 = x_1 - x_{min}$

$p_2 = \Delta x$, $q_2 = x_{max} - x_1$

$p_3 = -\Delta y$, $q_3 = y_1 - y_{min}$

$p_4 = \Delta y$, $q_4 = y_{max} - y_1$

Step 3: For each value of k from 1 to 4, perform the following check.

If $p_k = 0$ and $q_k < 0$ end the algorithm.

Step 4: For each value of k from 1 to 4 execute the following:

If $p_k < 0$ and $q_k/p_k > t_1$

then set $t_1 = q_k/p_k$

otherwise if $q_k/p_k < t_2$ then set $t_2 = q_k/p_k$.

Step 5: If $t_1 < t_2$, calculate $(x_i, y_i) = (x_1 + t_1\Delta x, y_1 + t_1\Delta y)$

Step 6: Draw the line from (x_i, y_i) to (x_f, y_f) where $(x_f, y_f) = (x_1 + t_2\Delta x, y_1 + t_2\Delta y)$.

PROGRAM:

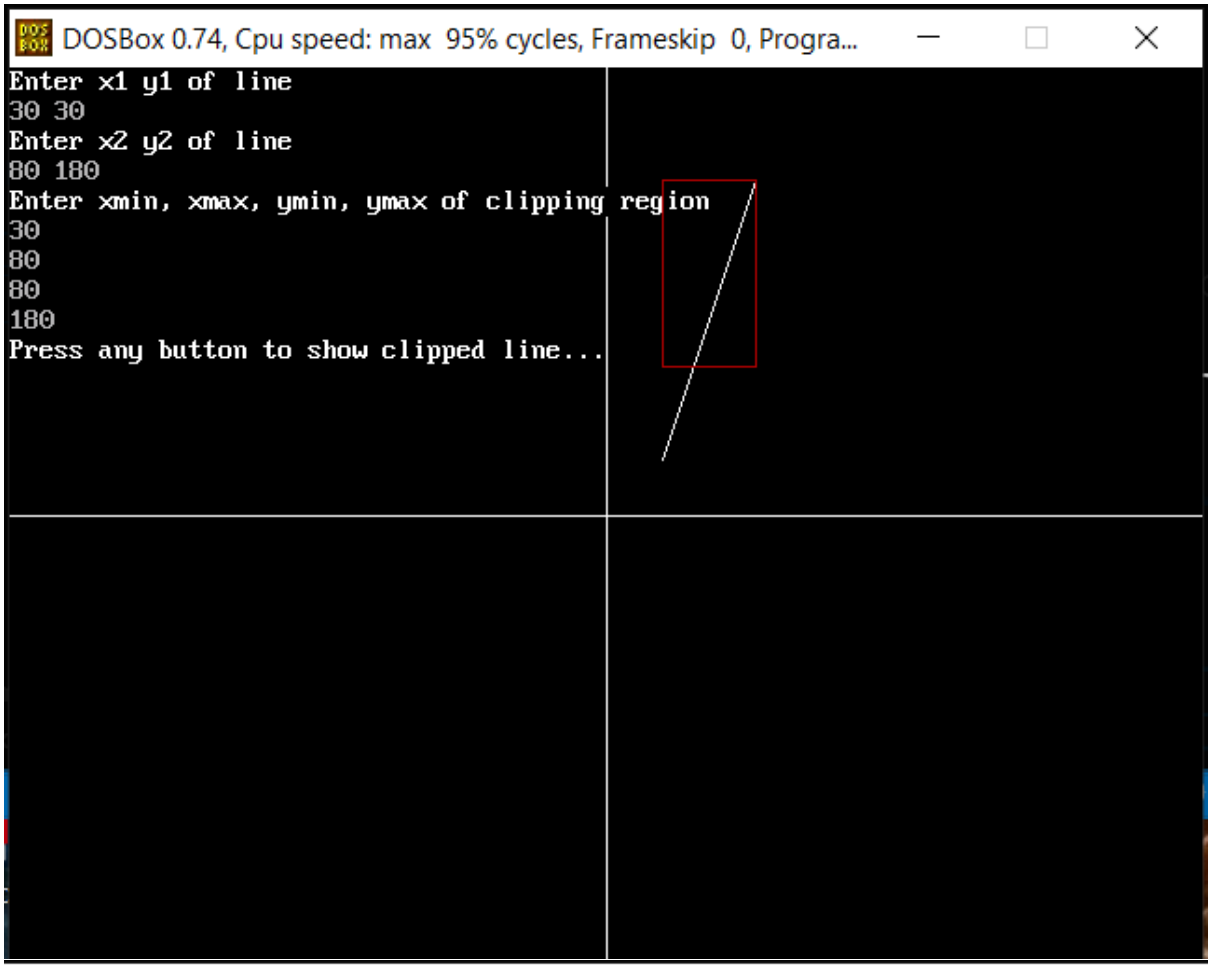
Code:

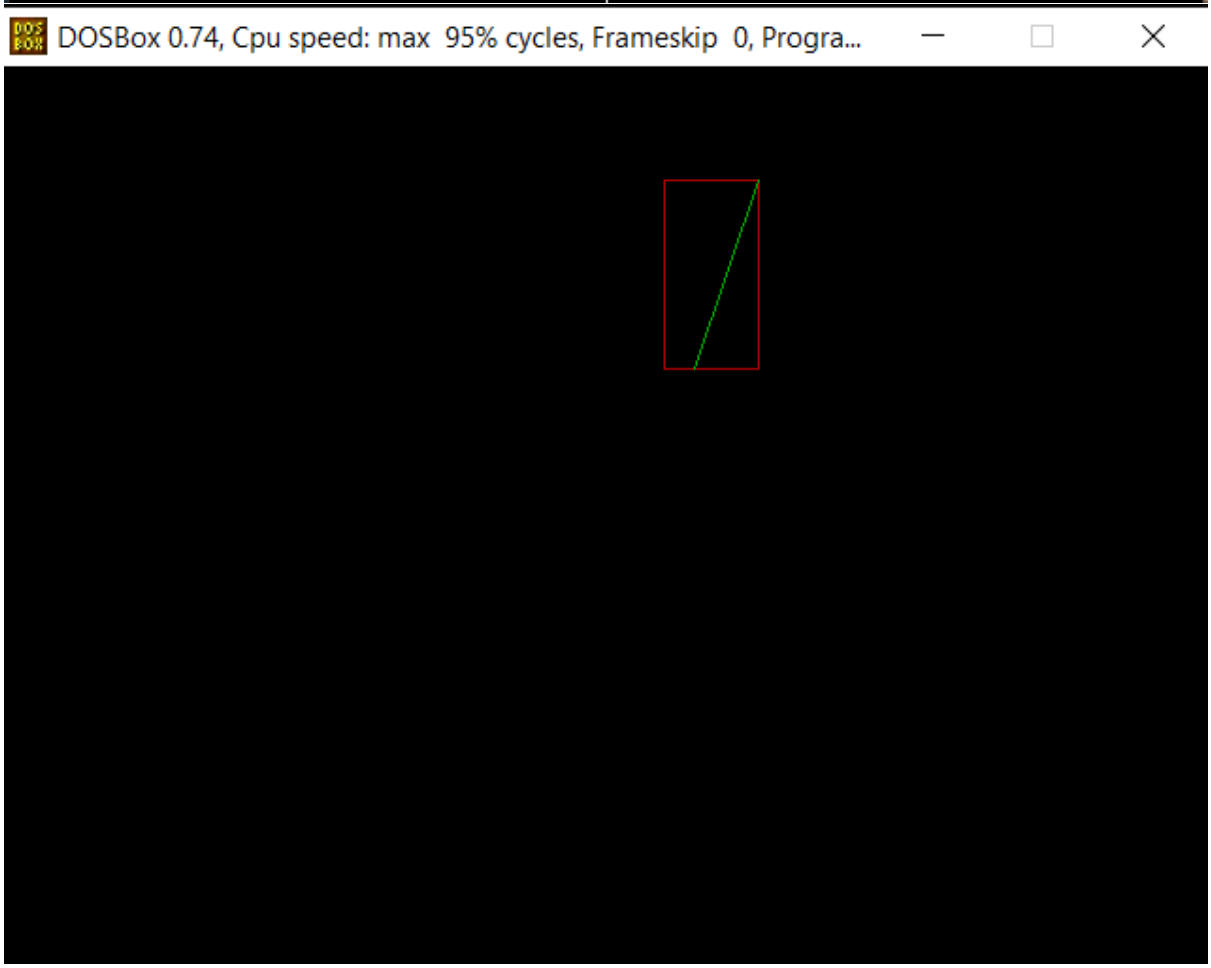
```
#include <stdio.h>
#include <conio.h>
#include <graphics.h>
void ClippedLine(int, int, int, int, int, int, int, int, int);
void Line(int, int, int, int);
void main()
{
   int x1, y1, x2, y2, xmin, xmax, ymin, ymax;
   int gd = DETECT, gm;
   initgraph(&gd, &gm, "C:\\TURBOC3\\BGI");
   line(320, 0, 320, 480);
   line(0, 240, 640, 240);
   printf("Enter x1 y1 of line\n");
   scanf("%d %d", &x1, &y1);
   printf("Enter x2 y2 of line\n");
   scanf("%d %d", &x2, &y2);
   Line(x1, y1, x2, y2);
   printf("Enter xmin, xmax, ymin, ymax of clipping region\n");
   scanf("%d %d %d %d", &xmin, &xmax, &ymin, &ymax);
   setcolor(RED);
   rectangle(320 + xmin, 240 - ymin, 320 + xmax, 240 - ymax);
   printf("Press any button to show clipped line...\n");
   getch();
   cleardevice();
   rectangle(320 + xmin, 240 - ymin, 320 + xmax, 240 - ymax);
```

```
    ClippedLine(x1, y1, x2, y2, xmin, ymin, xmax, ymax,
                GREEN);
    getch();
    getch();
    closegraph();
}
void ClippedLine(int x1, int y1, int x2, int y2, int xmin, int ymin, int xmax, int
ymax, int col)
{
    float t1, t2;
    int dx = x2 - x1, dy = y2 - y1, p[4], q[4], k;
    p[0] = -dx;
    p[1] = dx;
    p[2] = -dy;
    p[3] = dy;
    q[0] = x1 - xmin;
    q[1] = xmax - x1;
    q[2] = y1 - ymin;
    q[3] = ymax - y1;
    for (k = 0; k < 4; k++)
       if (p[k] == 0 && q[k] < 0)
          return;
    t1 = 0;
    t2 = 1;
    for (k = 0; k < 4; k++)
    {
       if (p[k] < 0 && (float)q[k] / p[k] > t1)
          t1 = (float)q[k] / p[k];
```

```
        else if (p[k] > 0 && (float)q[k] / p[k] < t2)
            t2 = (float)q[k] / p[k];
    }
    if (t1 < t2)
    {
        setcolor(col);
        Line(x1 + t1 * dx, y1 + t1 * dy, x1 + t2 * dx, y1 + t2 * dy);
    }
}
void Line(int x1, int y1, int x2, int y2)
{
    line(320 + x1, 240 - y1, 320 + x2, 240 - y2);
}
```

OUTPUT:


DOSBox 0.74, Cpu speed: max 95% cycles, Frameskip 0, Progra...
Enter x1 y1 of line
30 30
Enter x2 y2 of line
80 180
Enter xmin, xmax, ymin, ymax of clipping region
30
80
80
180
Press any button to show clipped line...


DOSBox 0.74, Cpu speed: max 95% cycles, Frameskip 0, Progra...