

CSI 422: Computer Graphics Lab

Assignment

Question 1

Problem Statement

Implement Sutherland-Hodgman polygon clipping algorithm to clip a polygon with respect to a clip rectangle formed by two vertical (left and right) and two horizontal (top and bottom) edges.

Input

- Four floating point numbers for four edges of clip rectangle: x_{\min} , x_{\max} , y_{\min} , y_{\max}
- Four characters for clipping order of four edges (L for Left, R for Right, T for Top, B for Bottom)
- No. of vertices in the polygon
- Co-ordinates (x and y values) of each vertex of the polygon. These co-ordinates will be floating point numbers.

Your program must handle file input according to the format given in sample input files.

Output

New sequence of vertex co-ordinates obtained after clipping with respect to each edge of the clip rectangle. After clipping with respect to an edge, generate a visual output displaying the original polygon and the latest clipped polygon. For each input, you will generate four visual outputs. See sample outputs for output format.

You can implement it using OpenGL libraries for C/C++.

Sample I/O

You are provided a sample *sutherland-hodgman.exe* and five sample input files in described format(*in-1.txt* to *in-5.txt*). The .exe program by default expects input from a file named *in.txt*. Rename any sample input file to *in.txt* or write your own input in a file named *in.txt*. Place the *in.txt* and the *sutherland-hodgman.exe* in same folder and then double click on the .exe to view generated output.

Output for

in-1.txt

```
Xmin = -50
Xmax = 150
Ymin = -100
Ymax = 200
Clipping edge sequence: L R B T
No. of vertices in the polygon = 3
P1 : (-100, -150)
P2 : (100, 250)
P3 : (200, -130)

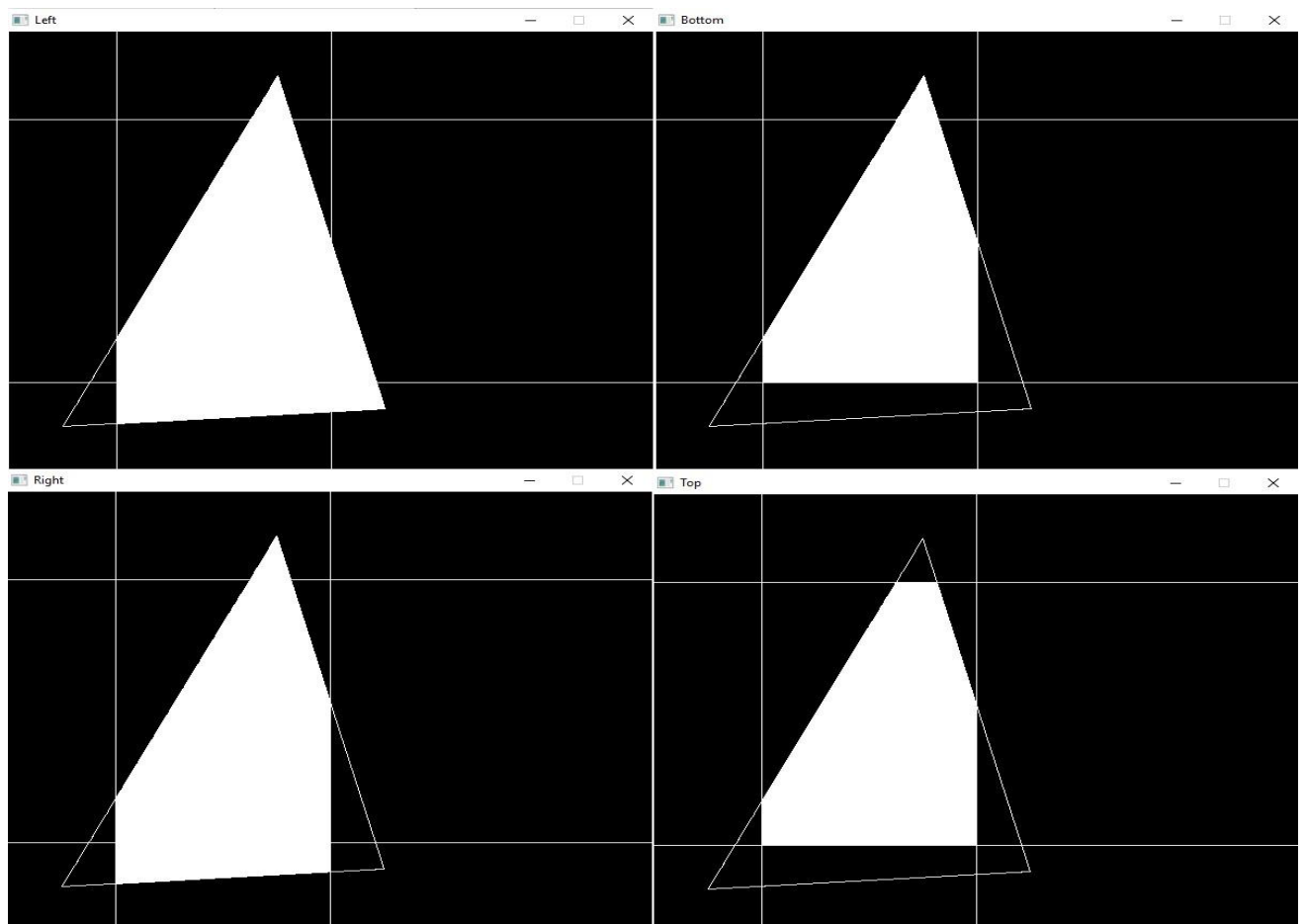
Clipping with respect to LEFT edge...
New Sequence : (-50, -50) (100, 250) (200, -130) (-50, -146.667)

Clipping with respect to RIGHT edge...
New Sequence : (100, 250) (150, 60) (150, -133.333) (-50, -146.667) (-50, -50)

Clipping with respect to BOTTOM edge...
New Sequence : (150, 60) (150, -100) (-50, -100) (-50, -50) (100, 250)

Clipping with respect to TOP edge...
New Sequence : (150, -100) (-50, -100) (-50, -50) (75, 200) (113.158, 200) (150, 60)

PRESS ANY KEY TO EXIT...
```



Output for

in-2.txt

```
Xmin = 100
Xmax = 300
Ymin = 100
Ymax = 300
Clipping edge sequence: L T B R
No. of vertices in the polygon = 9
P1 : (50, 150)
P2 : (200, 50)
P3 : (350, 150)
P4 : (260, 200)
P5 : (240, 340)
P6 : (200, 250)
P7 : (150, 350)
P8 : (80, 220)
P9 : (160, 180)

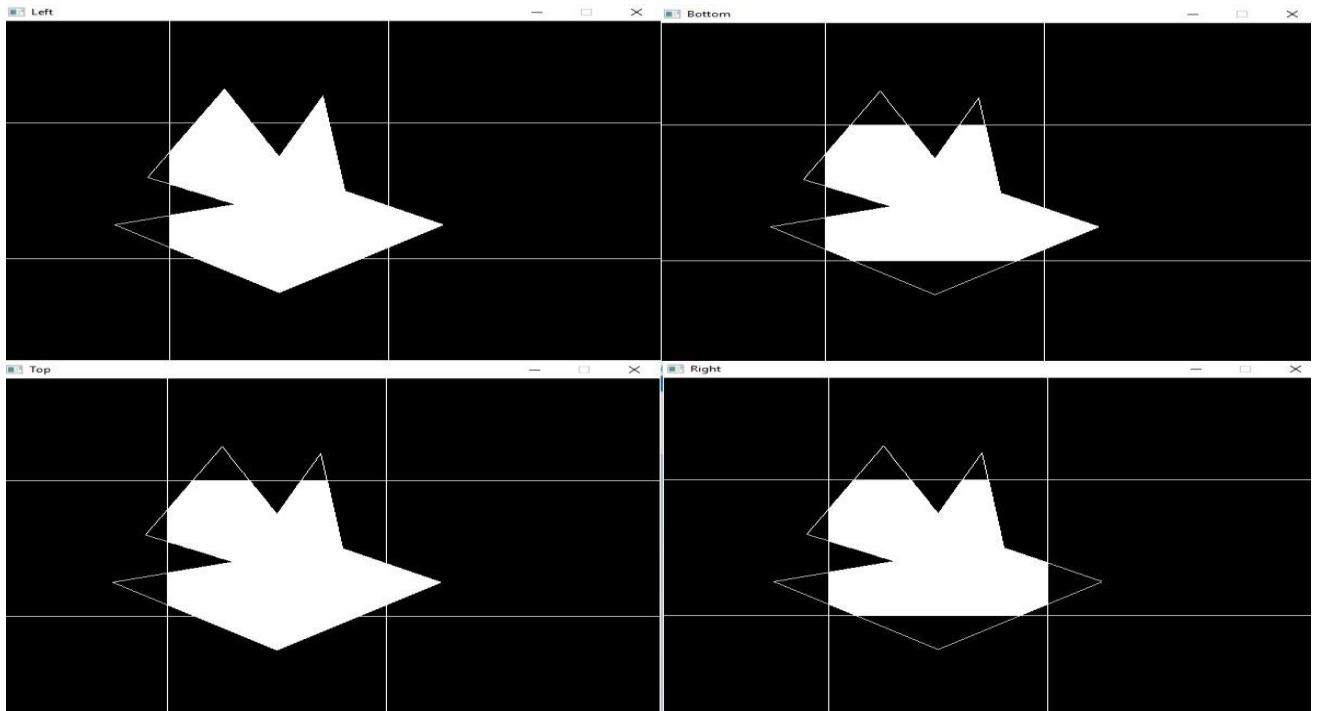
Clipping with respect to LEFT edge...
New Sequence : (100, 116.667) (200, 50) (350, 150) (260, 200) (240, 340) (200, 250) (150, 350) (100, 257.143) (100, 210) (160, 180) (100, 163.636)

Clipping with respect to TOP edge...
New Sequence : (200, 50) (350, 150) (260, 200) (245.714, 300) (222.222, 300) (200, 250) (175, 300) (123.077, 300) (100, 257.143) (100, 210) (160, 180) (100, 163.636) (100, 116.667)

Clipping with respect to BOTTOM edge...
New Sequence : (275, 100) (350, 150) (260, 200) (245.714, 300) (222.222, 300) (200, 250) (175, 300) (123.077, 300) (100, 257.143) (100, 210) (160, 180) (100, 163.636) (100, 116.667) (125, 100)

Clipping with respect to RIGHT edge...
New Sequence : (300, 116.667) (300, 177.778) (260, 200) (245.714, 300) (222.222, 300) (200, 250) (175, 300) (123.077, 300) (100, 257.143) (100, 210) (160, 180) (100, 163.636) (100, 116.667) (125, 100) (275, 100)

PRESS ANY KEY TO EXIT...
```



Output for

in-3.txt

```
Xmin = 150
Xmax = 200
Ymin = 150
Ymax = 200
Clipping edge sequence: T L R B
No. of vertices in the polygon = 3
P1 : (150, 150)
P2 : (200, 150)
P3 : (175, 250)

Clipping with respect to TOP edge...
New Sequence : (200, 150) (187.5, 200) (162.5, 200) (150, 150)

Clipping with respect to LEFT edge...
New Sequence : (187.5, 200) (162.5, 200) (150, 150) (200, 150)

Clipping with respect to RIGHT edge...
New Sequence : (162.5, 200) (150, 150) (200, 150) (187.5, 200)

Clipping with respect to BOTTOM edge...
New Sequence : (150, 150) (200, 150) (187.5, 200) (162.5, 200)

PRESS ANY KEY TO EXIT...
```

Output for *in-4.txt*

```
Xmin = -30
Xmax = 30
Ymin = -40
Ymax = 40
Clipping edge sequence: T B L R
No. of vertices in the polygon = 4
P1 : (-50, -70)
P2 : (-90, 20)
P3 : (-10, 60)
P4 : (0, 0)

Clipping with respect to TOP edge...
New Sequence : (-90, 20) (-50, 40) (-6.66667, 40) (0, 0) (-50, -70)

Clipping with respect to BOTTOM edge...
New Sequence : (-50, 40) (-6.66667, 40) (0, 0) (-28.5714, -40) (-63.3333, -40) (-90, 20)

Clipping with respect to LEFT edge...
New Sequence : (-30, 40) (-6.66667, 40) (0, 0) (-28.5714, -40) (-30, -40)

Clipping with respect to RIGHT edge...
New Sequence : (-6.66667, 40) (0, 0) (-28.5714, -40) (-30, -40) (-30, 40)

PRESS ANY KEY TO EXIT...
```

Output for

in-5.txt

```
Xmin = 40
Xmax = 80
Ymin = 30
Ymax = 100
Clipping edge sequence: R L T B
No. of vertices in the polygon = 3
P1 : (40, 100)
P2 : (100, 30)
P3 : (20, 30)

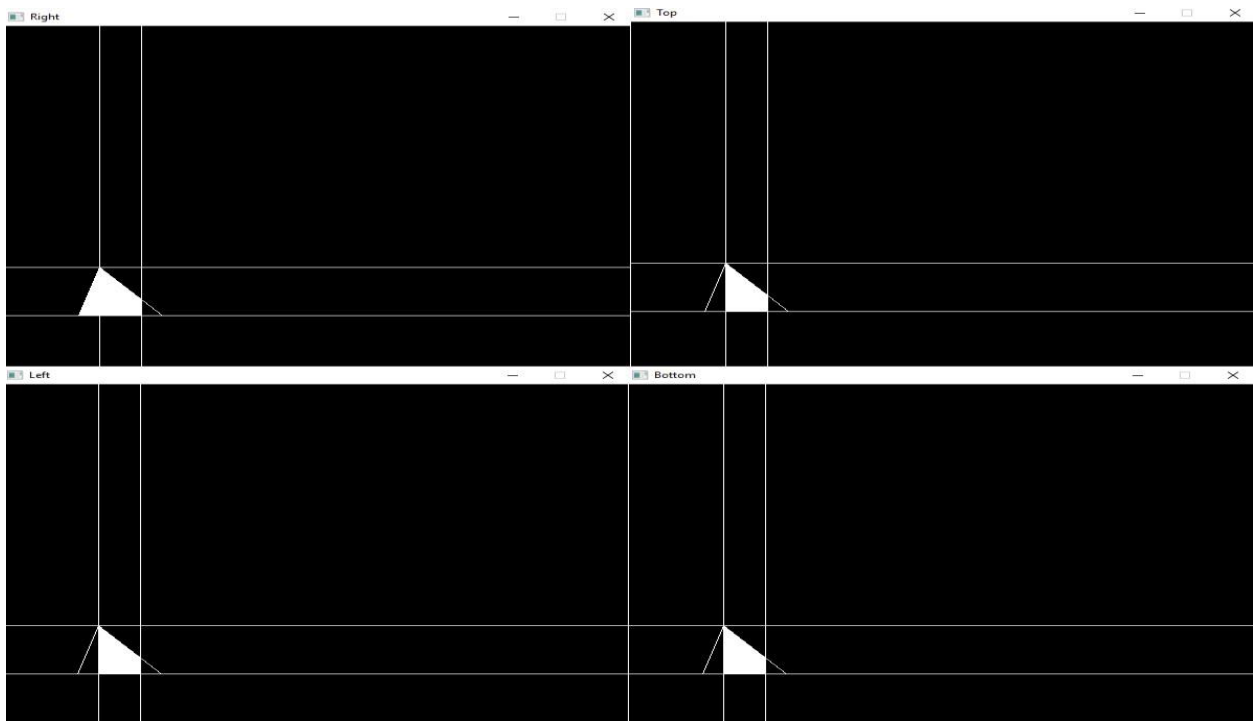
Clipping with respect to RIGHT edge...
New Sequence : (80, 53.3333) (80, 30) (20, 30) (40, 100)

Clipping with respect to LEFT edge...
New Sequence : (80, 30) (40, 30) (40, 100) (40, 100) (80, 53.3333)

Clipping with respect to TOP edge...
New Sequence : (40, 30) (40, 100) (40, 100) (80, 53.3333) (80, 30)

Clipping with respect to BOTTOM edge...
New Sequence : (40, 100) (40, 100) (80, 53.3333) (80, 30) (40, 30)

PRESS ANY KEY TO EXIT...
```



Special Case

Consider the polygon abc of *in-5.txt*. Vertices a , b and c fall on edges of the clip rectangle. In cases like this, where a polygon vertex fall on some edge of clip rectangle, clipped output may contain duplicate vertices. Notice the new sequences after clipping with respect to left, top and bottom edges duplicate copies of a .

i. Right

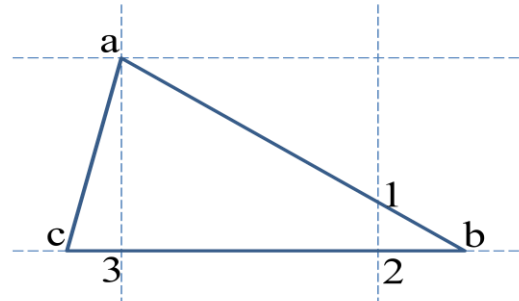
ab in-out 1
bc out-in 2c
ca in-in a

New: 12ca

ii. Left

12 in-in 2
2c in-out 3
ca out-in aa
a1 in-in 1

New: 23aa1



iii. Top

23 in-in 3
3a in-in a
aa in-in a
a1 in-in 1
12 in-in 2

New: 3aa12

iv. Bottom

3a in-in a
aa in-in a
a1 in-in 1
12 in-in 2
23 in-in 3

New: aa123

In your implementation, handle the above mentioned scenario so that duplicate vertices are not passed to next stage. For example, if left clipping produces 23aa1, pass 23a1 to top clipping; if top clipping produces 3aa12, pass 3a12 to bottom clipping, if bottom clipping produces aa123, present a123 as final output.

N.B: Given sample *sutherland-hodgman.exe* does not handle this duplicate vertex scenario.

Implementation Overview

You may follow the following structure in your program. Of course, maintaining this structure is completely optional.

```

// keep two list of vertices, one list prev for the old sequence, another list next for the
updated sequence

// ensure that while clipping with respect to each edge, the sequence resulted from the
previous clipping is in prev and the sequence after the current clipping is in next
left_clip(){
    iterate the vertex sequence from prev and in each iteration add output
    vertices (according to the Sutherland-Hodgman clipping rules) in the list next
}
right_clip(){
    similar as left clip
}
top_clip(){ similar as
left clip
}
bottom_clip(){
    similar as left clip
}
sutherland_hodgman(){
    for i = 1 to 4
        check which edge comes at ith position of clipping order and call
        the corresponding clip function          print the updated vertex
sequence from next          copy the sequence from next to prev
    }
main() { take input from file
sutherland_hodgman()
}

```

Question 2

Implement any other line clipping or polygon clipping or 3D clipping algorithm in same manner.

Deadline: June 8, 2022 (Wednesday)

Marks distribution:

Algorithm	Algorithmic computation	I/O	OpenGL visualization	Viva	Total
Algorithm 1 (sutherland-hodgman)	1.5	0.5	2.5	2.5	7
Algorithm 2	2.5	0.5	2.5	2.5	8
Total					15