|      | 0 |
|------|---|
| Date |   |
| Page |   |
| 1    |   |

## Assignment No-A-5

Student Name: - Sameer Many Brambecha.

ROLL NO: - 21115

Batch: - E-1

Performance Date: 19-10-2021

Submission Date: 12-10-2021.

Title: > Cohen-Sutherland Line Clipping

Exoblem Statement: -> Claite a C++ program to implement Cohen Sutherland line clipping algorithm.

Learning Objectives: >

- 1) To understand the Cohen-Sutherland line clipping
- 2) To understand the use of mouse interfacing.

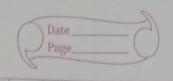
hearning Outcomes: -> After completion of this assignments, students will be able to implement Cohen-Sutherland line clipping algorithm.

S/W & M/W requirement:-

1) 64-bit open source hinux.

2:> Open Source C++ programming tool like G++/4CC

8.7 Open GL.



## Theory: >

cohen Sutherland algorithm is one of the popular line alipping algorithm. To speed up the prousely, this algorithm performs initial tests that reduce the number of infersections that must be calculated. This algorithm uses a four digit code to indicate which of nine regions contain the end point of line. The four bit codes are called region codes or out codes. These codes identity the speation of the point relative to the boundaries of the clipping trustangle. Fach bits position in the region code is used to indicate one of the four relative coordinate positions of the point with respective to the clipping window to the left right, top or bottom. The bits are set to I based on the following of theme.

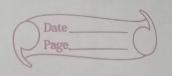
set bit 1: if the end points is to the left of the window.

Set bit 2: if the end points is to the right of the window.

Set bit 3: if the end points is below the window.

Set bit 4: if the end points is below the window.

| 1010           | 1000             | 1001       |
|----------------|------------------|------------|
| region 1       | region 2         | region 3   |
|                | the contract has | Jan Alland |
| a manda a cont | 0000             | 2001       |
| 0010           | Clipping Window  | 0001       |
| region.        |                  | region 4   |
| 0              |                  | 1.01       |
| 0110           | 0100             | 0101       |
| region 7       | region 6         | region 5   |
| 0              |                  |            |



The equation for line passing through points pi(x, 4,) and fr (x, 42) is

y=m(x-x,)+y, or u=m(x-x,)+y,

y=m(x-22)+/2

where m = 42-41 (Slope of a line)

Therefore, the intersections with the dipping boundaries of the window are given as:

deft: xLzy=m(xl-x1)+y, Right: xR, y=m(xR-x1)+y,

Top: yT, x = x, + (1/m) (yT-yi)
Bottom yBz, x = x, + (1/m) (yB-yi)

Algorithm:

1.) Read two endpoints of the time say pi(x1,4) and p2 (x2, 42).

a) Read two corners (left-top & right bottom) of the window, vay (wx1, wy, wx2, wy2).

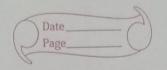
8.) Assign the region codes for two endpoints p12 p2 using following steps:-

set bitt - if (x<wx)

set bite - if (gar waz)

set bit3 - If (y < wy2)

SA bit 4 - if (y' wy1)



4) Check for visibility of line pipe

zero then the line is completely visible. Hence draw the line and go to step. 9.

b) if region codes for endpoints are not zero and the logical Anding of them is also non-zero then the line is completely invisible, so reject the line and go to step 9.

the conditions in (4a) and (4b), then the line is partially visible.

5) Determine the intersecting edge of the alipping window by inspecting the segion codes of two endpoints

a) if region codes for both the end points are non-zero, sind intosection points pi and p2' with boundary edges of dipping window with respect to point p1 and point p2, respectively.

b) if region code for any one end point is non-zero then find intersection point pi'ar p2' with the boundary selge of the clipping window with respect to it.

6) Divide the line signents considering intersection points.

7) Reject the line segment if any one end point of if appears outsides the dipping window.

8.) Dears the remaining line segments.

9) Stap. Stop -

