

## **Experiment no : 02**

Name : Atharva B. Iparkar

Date : 18/09/2024

Roll no : S211045

Class : S.E.

Div : A

Batch : A-2

### **Problem Statement :**

Write a C++ program to implement Cohen Southerland line clipping algorithm

### **Code :**

```
#include<iostream>
```

```
#include<graphics.h>
```

```
using namespace std ;
```

```
class Coordinate {
```

```
public :
```

```
int x,y;
```

```
char code[4];
```

```
};
```

```
class Lineclip {
```

```
public :
```

```
Coordinate p;
```

```
void drawwindow() {  
  
    line(150,100,450,100);  
  
    line(450,100,450,350);  
  
    line(450,350,150,350);  
  
    line(150,350,150,100);  
  
}
```

```
void drawline(Coordinate p1,Coordinate p2) {  
  
    line(p1.x,p1.y,p2.x,p2.y);  
  
}
```

```
Coordinate setcode(Coordinate p) {  
  
    Coordinate ptemp;  
  
    if(p.y<100)  
  
        ptemp.code[0]='1';  
  
    else  
  
        ptemp.code[0]='0';  
  
    if(p.y>350)  
  
        ptemp.code[1]='1';  
  
    else  
  
        ptemp.code[1]='0';  
  
    if(p.x>450)
```

```
ptemp.code[2]='1';
```

```
else
```

```
ptemp.code[2]='0';
```

```
if(p.x<150)
```

```
ptemp.code[3]='1';
```

```
else
```

```
ptemp.code[3]='0';
```

```
ptemp.x=p.x;
```

```
ptemp.y=p.y;
```

```
return(ptemp);
```

```
}
```

```
int visibility(Coordinate p1,Coordinate p2) {
```

```
int i,flag=0;
```

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

```
if((p1.code[i]!='0') || (p2.code[i]!='0'))
```

```
flag=1;
```

```
}
```

```
if(flag==0)
```

```
return(0);
```

```
for(i=0;i<4;i++) {  
  
if((p1.code[i]==p2.code[i]) && (p1.code[i]=='1'))  
  
flag='0';  
  
}
```

```
if(flag==0)  
  
return(1);  
  
return(2);  
  
}
```

```
Coordinate resetendpt(Coordinate p1,Coordinate p2) {  
  
Coordinate temp;  
  
int x,y,i;  
  
float m,k;  
  
  
if(p1.code[3]=='1')  
  
x=150;  
  
if(p1.code[2]=='1')  
  
x=450;  
  
if((p1.code[3]=='1') || (p1.code[2]=='1')) {  
  
m=(float)(p2.y-p1.y)/(p2.x-p1.x);
```

```
k=(p1.y+(m*(x-p1.x)));
```

```
temp.y=k;
```

```
temp.x=x;
```

```
for(i=0;i<4;i++)
```

```
temp.code[i]=p1.code[i];
```

```
if(temp.y<=350 && temp.y>=100)
```

```
return (temp);
```

```
}
```

```
if(p1.code[0]=='1')
```

```
y=100;
```

```
if(p1.code[1]=='1')
```

```
y=350;
```

```
if((p1.code[0]=='1') || (p1.code[1]=='1')) {
```

```
m=(float)(p2.y-p1.y)/(p2.x-p1.x);
```

```
k=(float)p1.x+(float)(y-p1.y)/m;
```

```
temp.x=k;
```

```
temp.y=y;
```

```
for(i=0;i<4;i++)

temp.code[i]=p1.code[i];

return(temp);

} else

return(p1);

}

};
```

```
int main() {

class Lineclip l ;

int gd=DETECT,v,gm;

Coordinate p1,p2,p3,p4,ptemp;

cout<<"\nEnter x1 and y1 : \n";

cin>>p1.x>>p1.y;

cout<<"\nEnter x2 and y2 : \n";

cin>>p2.x>>p2.y;

initgraph(&gd,&gm,NULL);

l.drawwindow();

delay(5000);

l.drawline(p1,p2);

delay(5000);

cleardevice();
```

```
delay(5000);

p1=l.setcode(p1);

p2=l.setcode(p2);

v=l.visibility(p1,p2);

cout<<v;

delay(5000);

switch(v) {

case 0:

l.drawwindow();

delay(5000);

l.drawline(p1,p2);

break;


case 1: l.drawwindow();

delay(5000);

break;


case 2: p3=l.resetendpt(p1,p2);

p4=l.resetendpt(p2,p1);

l.drawwindow();

delay(5000);

l.drawline(p3,p4);
```

```
break;
```

```
}
```

```
delay(5000);
```

```
closegraph();
```

```
}
```

Output :

```
d_comp_pl_ii_11@d-comp-pl-ii-11:~/SE_A2_S211045_Atharva$ g++ LineClipping.cpp -o l -  
lgraph
```

```
d_comp_pl_ii_11@d-comp-pl-ii-11:~/SE_A2_S211045_Atharva$ ./l
```

Enter x1 and y1

100

200

Enter x2 and y2

300

400

[xcb] Unknown sequence number while processing queue

[xcb] Most likely this is a multi-threaded client and XInitThreads has not been called

[xcb] Aborting, sorry about that.

l: ../../src/xcb\_io.c:260: poll\_for\_event: Assertion '!xcb\_xlib\_threads\_sequence\_lost' failed.





