DATE: 06/09/2023

GROUP A: PRATICAL.2

NAME: SAYALI TANAJI PAWAR
ROLL NO: S213002
CLASS: SE
DIV: C
BATCH: C1

PROBLEM STATEMENT:

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

CODE:

```
#include<iostream>
#include<stdlib.h>
#include<math.h>
#include<graphics.h>
//#include<dos.h>
using namespace std;
class Coordinate                    //creating a class for declaring coordinates
{
public:
int x,y;
char code[4];
};
class Lineclip                      //class for declaring all the functions required
{
public:
Coordinate PT;                          //creating object of class coordinate
void drawwindow();
void drawline(Coordinate p1,Coordinate p2);
Coordinate setcode(Coordinate p);
int visibility(Coordinate p1,Coordinate p2);
Coordinate resetendpt(Coordinate p1,Coordinate p2);
};
int main()
{
Lineclip lc;                                //object for class lineclipping
```

```
int gd = DETECT,v,gm;
Coordinate p1,p2,p3,p4,ptemp;
cout<<"\n Enter x1 and y1\n";
cin>>p1.x>>p1.y;
cout<<"\n Enter x2 and y2\n";
cin>>p2.x>>p2.y;
initgraph(&gd,&gm,NULL);                //initialization of graphics
lc.drawwindow();
delay(20000);
lc.drawline (p1,p2);
delay(20000);
cleardevice();
delay(20000);
p1=lc.setcode(p1);
p2=lc.setcode(p2);
v=lc.visibility(p1,p2);
delay(200000);
switch(v)                               //using switch case for user choice
{
case 0: lc.drawwindow();
delay(20000);
lc.drawline(p1,p2);
break;
case 1:lc.drawwindow();
delay(20000);
break;
case 2:p3=lc.resetendpt(p1,p2);
p4=lc.resetendpt(p2,p1);
lc.drawwindow();
delay(20000);
lc.drawline(p3,p4);
break;
}
delay(20000);
closegraph();
}
void Lineclip::drawwindow()                     //function for creating
{
line(150,100,450,100);
line(450,100,450,350);
line(450,350,150,350);
line(150,350,150,100);
}
void Lineclip::drawline(Coordinate p1,Coordinate p2)   //drawing line used for
clipping
```

```
{
line(p1.x,p1.y,p2.x,p2.y);
}
Coordinate Lineclip::setcode(Coordinate p)     //creating the region code
{
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 Lineclip:: visibility(Coordinate p1,Coordinate p2)   //checking visibility of line
{
int i,flag=0;
for(i=0;i<4;i++)
{
if(p1.code[i]!='0' || (p2.code[i]=='1'))
flag='0';
}
if(flag==0)
return(0);                                          //switch choice 0
for(i=0;i<4;i++)
{
if(p1.code[i]==p2.code[i] && (p2.code[i]=='1'))
flag='0';
}
if(flag==0)
return(1);                                           //switch choice 1
return(2);                                          //switch choice 2
}
```

```
Coordinate Lineclip::resetendpt(Coordinate p1,Coordinate p2)   //reseting points after clipping
{
Coordinate temp;
int x,y,i;
float m,k;
if(p1.code[3]=='1')
x=150;                                          //xmin
if(p1.code[2]=='1')
x=450;                                          //xmax
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;                                          //ymin
if(p1.code[1]=='1')
y=350;                                              //ymax
if((p1.code[1]=='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);
}
```

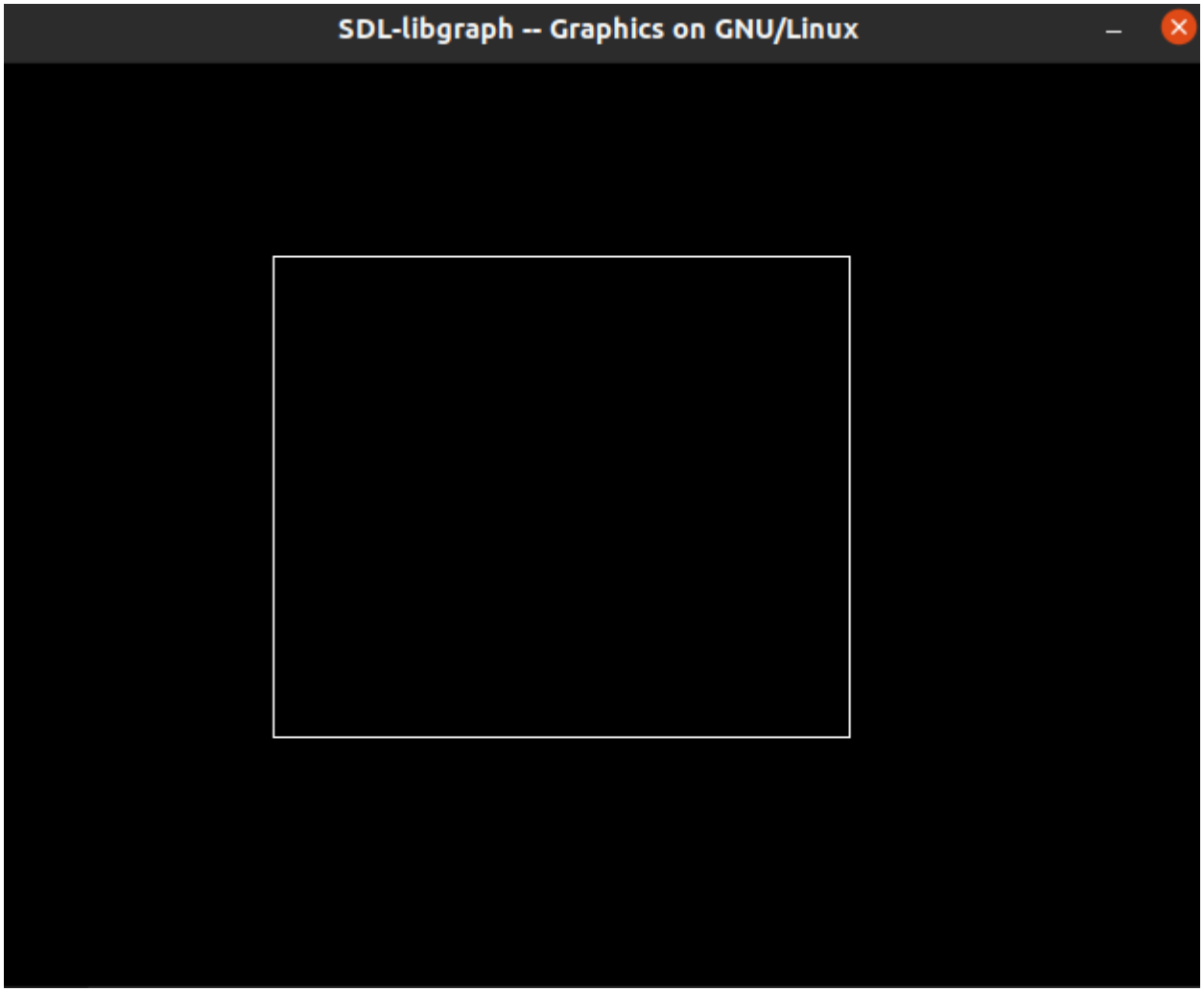
INPUT:

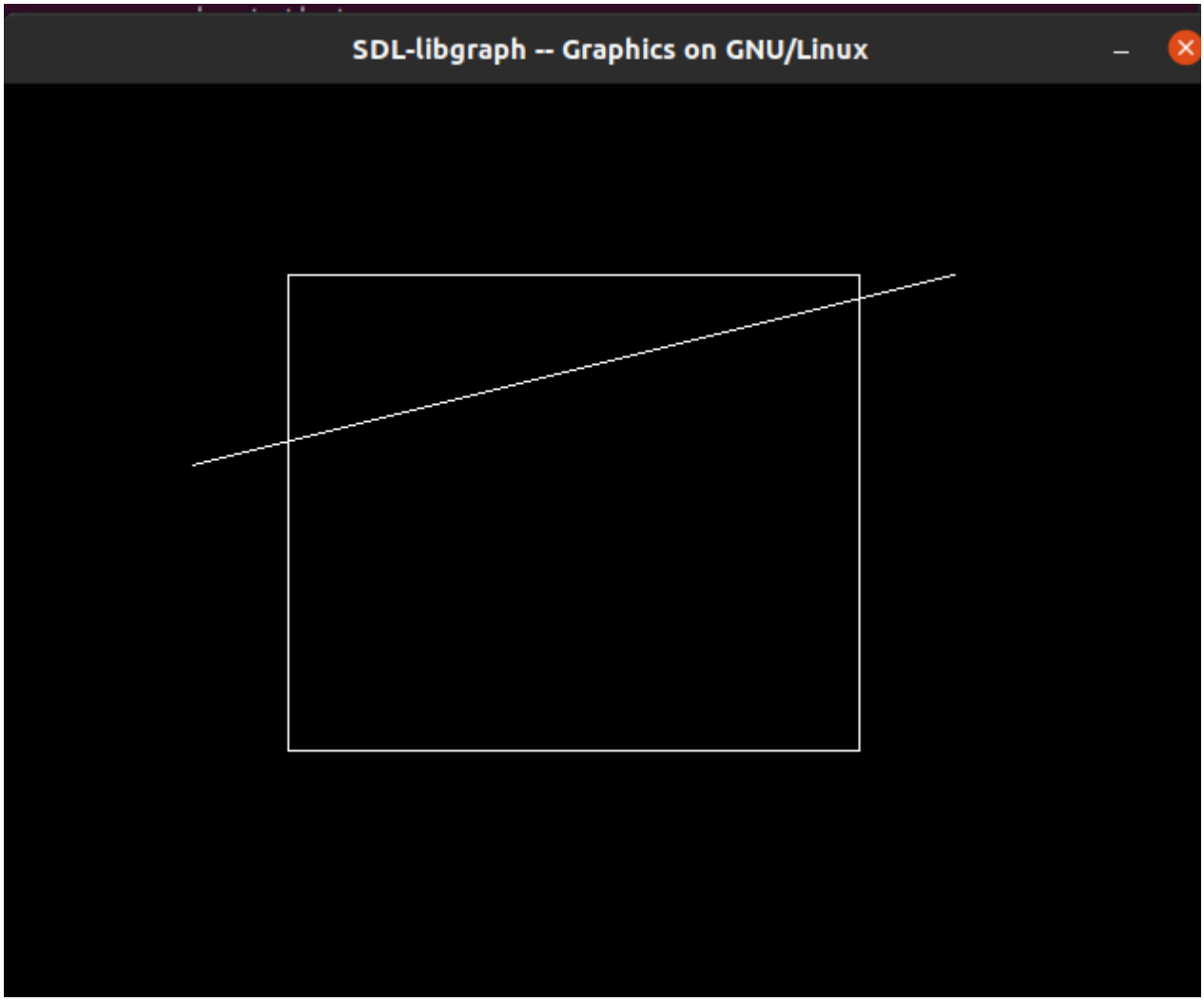
```
d_comp_sli_02@d-comp-sli-02:~$ g++ cg.cpp -o abc -lgraph
d_comp_sli_02@d-comp-sli-02:~$ ./abc
```

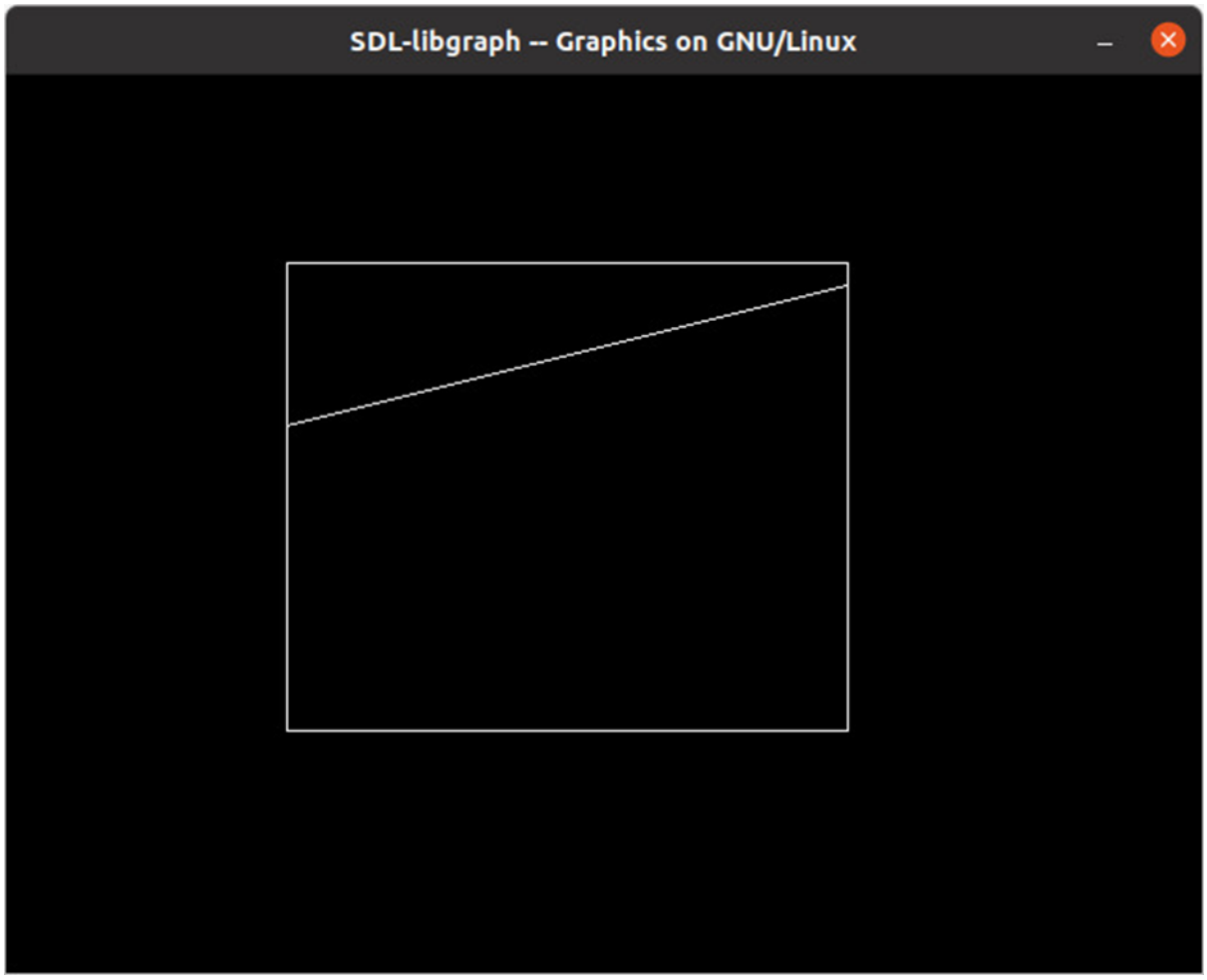
Enter x1 and y1
100
200

Enter x2 and y2
500
100
[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.
abc: ../../src/xcb_io.c:260: poll_for_event: Assertion
`!xcb_xlib_threads_sequence_lost' failed.

OUTPUT:




SDL-libgraph -- Graphics on GNU/Linux


SDL-libgraph -- Graphics on GNU/Linux