

COMPUTER GRAPHICS

(CA3304)

NIT Jamshedpur



SESSION 2021-2022

Name: Vishakha Barahdia

Roll no.: 2020PGCACA96

Course: MCA 3rd semester

Subject: Computer Graphics Assignment 03

Date: 15/11/2021

Faculty-in-charge: Dr. Mudassir Rafi

1. Write a C-program for performing clipping operations on line entered by user using Cohen Sutherland line clipping algorithm.

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
#include<graphics.h>
#include<dos.h>

typedef struct coordinate
{
int x,y;
char code[4];
}PT;
void drawwindow();
void drawline(PT p1,PT p2);
PT setcode(PT p);
int visibility(PT p1,PT p2);
PT resetendpt(PT p1,PT p2);

int main()
{
int gd=DETECT,v,gm;
PT p1,p2,p3,p4,ptemp;
printf("\nEnter x1 and y1\n");
scanf("%d %d",&p1.x,&p1.y);
```

```
printf("\nEnter x2 and y2\n");
scanf("%d %d",&p2.x,&p2.y);
initgraph(&gd,&gm,"c:\\turboc3\\bgi");
drawwindow();
delay(5000);
drawline(p1,p2);
delay(5000);
cleardevice();
delay(5000);
p1=setcode(p1);
p2=setcode(p2);
v=visibility(p1,p2);
delay(5000);
switch(v)
{
case 0: drawwindow();
delay(5000);
drawline(p1,p2);

break;
case 1: drawwindow();
delay(5000);
break;
case 2: p3=resetendpt(p1,p2);
p4=resetendpt(p2,p1);
drawwindow();
delay(5000);
drawline(p3,p4);
break;
}
delay(35000);
closegraph();
}
```

```

void drawwindow()
{
    line(150,100,450,100);
    line(450,100,450,350);
    line(450,350,150,350);
    line(150,350,150,100);
}
void drawline(PT p1,PT p2)
{
    line(p1.x,p1.y,p2.x,p2.y);
}
PT setcode(PT p) //for setting the 4 bit code
{
    PT ptemp;
    if(p.y<100)
        ptemp.code[0]='1'; //Top
    else
        ptemp.code[0]='0';
    if(p.y>350)
        ptemp.code[1]='1'; //Bottom
    else
        ptemp.code[1]='0';
    if(p.x>450)
        ptemp.code[2]='1'; //Right
    else
        ptemp.code[2]='0';
    if(p.x<150)
        ptemp.code[3]='1'; //Left
    else
        ptemp.code[3]='0';
    ptemp.x=p.x;
    ptemp.y=p.y;
    return(ptemp);
}

```

```
}
```

```
int visibility(PT p1,PT 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);
}

PT resetendpt(PT p1,PT p2)
{
    PT 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);
```

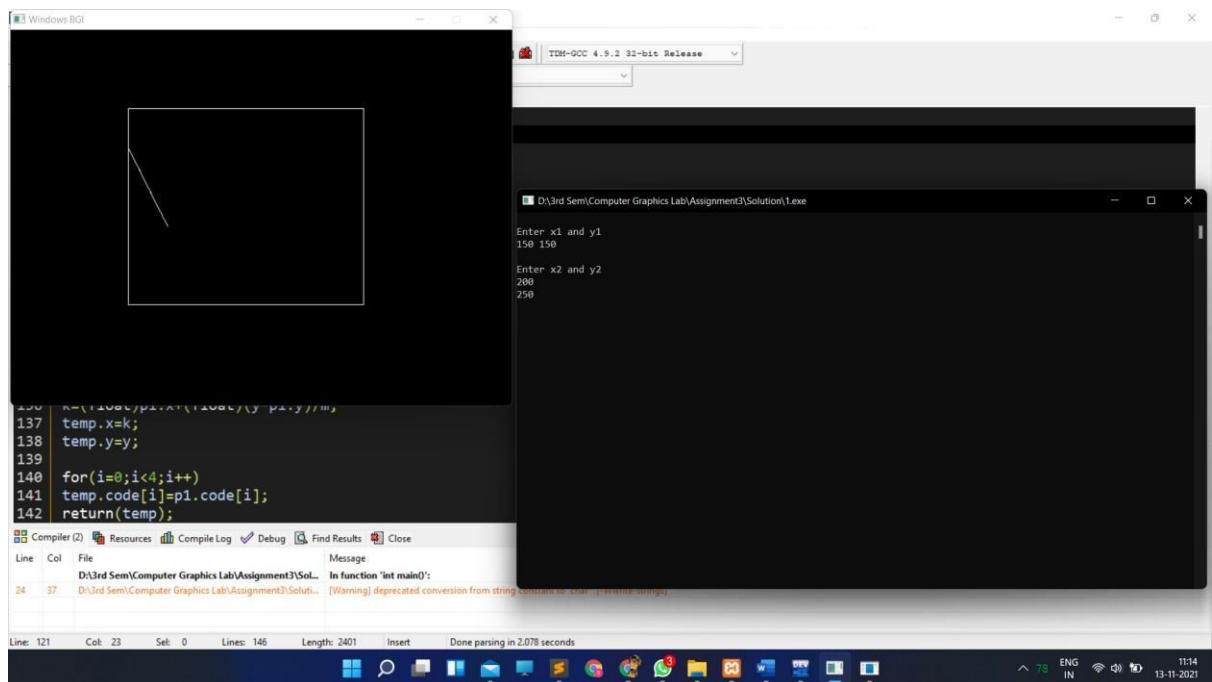
```

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);
}

```

OUTPUT:



2. Write a C program to clip a polygon entered by a user using Sutherland Hodgeman clipping algorithm

```
#include<stdio.h>
#include<graphics.h>
#include<conio.h>
#include<stdlib.h>
int main()
{
    int gd, gm, n, *x, i, k=0;
    //window coordinates int
    wx1=220, wy1=140, wx2=420, wy2=140, wx3=420, wy3=340, wx4=220, wy4=340;
    int w[]={220,140,420,140,420,340,220,340,220,140}; //array for drawing window
```

```

detectgraph(&gd,&gm);
initgraph(&gd,&gm,"c:\\turbo3\\bgi"); //initializing graphics
printf("Window:-");
setcolor(RED); //red colored window
drawpoly(5,w); //window drawn
printf("Enter the no. of vertices of polygon: ");
scanf("%d",&n);
// x = malloc(n*2+1);
printf("Enter the coordinates of points:\n");
k=0;
for(i=0;i<n*2;i+=2) //reading vertices of polygon
{
printf("(x%d,y%d): ",k,k);
scanf("%d,%d",&x[i],&x[i+1]);
k++;
}
x[n*2]=x[0]; //assigning the coordinates of first vertex to last
additional vertex for drawpoly method.
x[n*2+1]=x[1];
setcolor(WHITE);
drawpoly(n+1,x);
printf("\nPress a button to clip a polygon..");
getch();
setcolor(RED);
drawpoly(5,w);
setfillstyle(SOLID_FILL,BLACK);
floodfill(2,2,RED);
// gotoxy(1,1); //bringing cursor at starting position
printf("\nThis is the clipped polygon..");
getch();

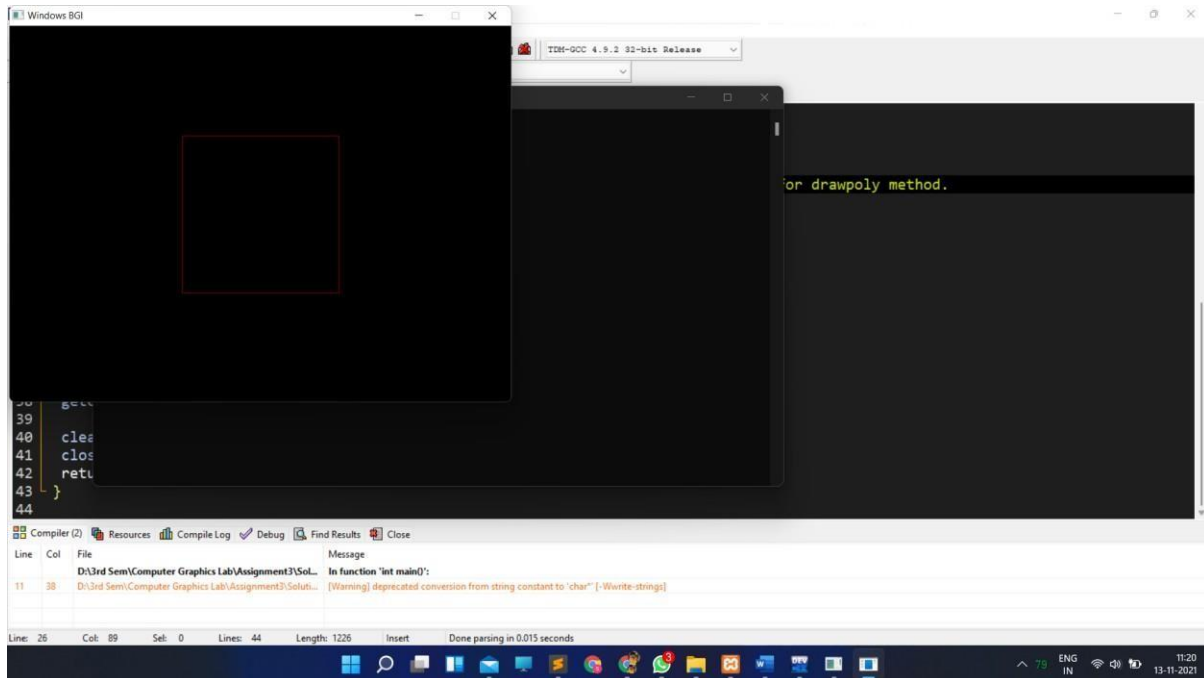
cleardevice();
closegraph();

```



```
return 0;  
}
```

OUTPUT:



3. Write a c program to perform shear operation on a cuboid object. Ask the shear parameters from user and accordingly display the shear object.

```
#include<stdio.h>  
#include<graphics.h>  
#include<conio.h>  
int main()  
{
```

```
int gd=DETECT,gm;

int x,y,x1,y1,x2,y2,x3,y3,shear_f;
initgraph(&gd,&gm,"C:\\TURBOC3\\BGI");
printf("\n please enter first coordinate = ");
scanf("%d %d",&x,&y);
printf("\n please enter second coordinate = ");
scanf("%d %d",&x1,&y1);
printf("\n please enter third coordinate = ");
scanf("%d %d",&x2,&y2);
printf("\n please enter last coordinate = ");
scanf("%d %d",&x3,&y3);
printf("\n please enter shearing factor y = ");
scanf("%d",&shear_f);
```

```
cleardevice();
line(x,y,x1,y1);
line(x1,y1,x2,y2);

line(x2,y2,x3,y3);

line(x3,y3,x,y);
setcolor(RED);
y=y+ x*shear_f;
y1=y1+ x1*shear_f;
y2=y2+ x2*shear_f;
y3=y3+ x3*shear_f;
line(x,y,x1,y1);
line(x1,y1,x2,y2);
line(x2,y2,x3,y3);

line(x3,y3,x,y);
getch();
closegraph();
return 0;
}
```

OUTPUT:

