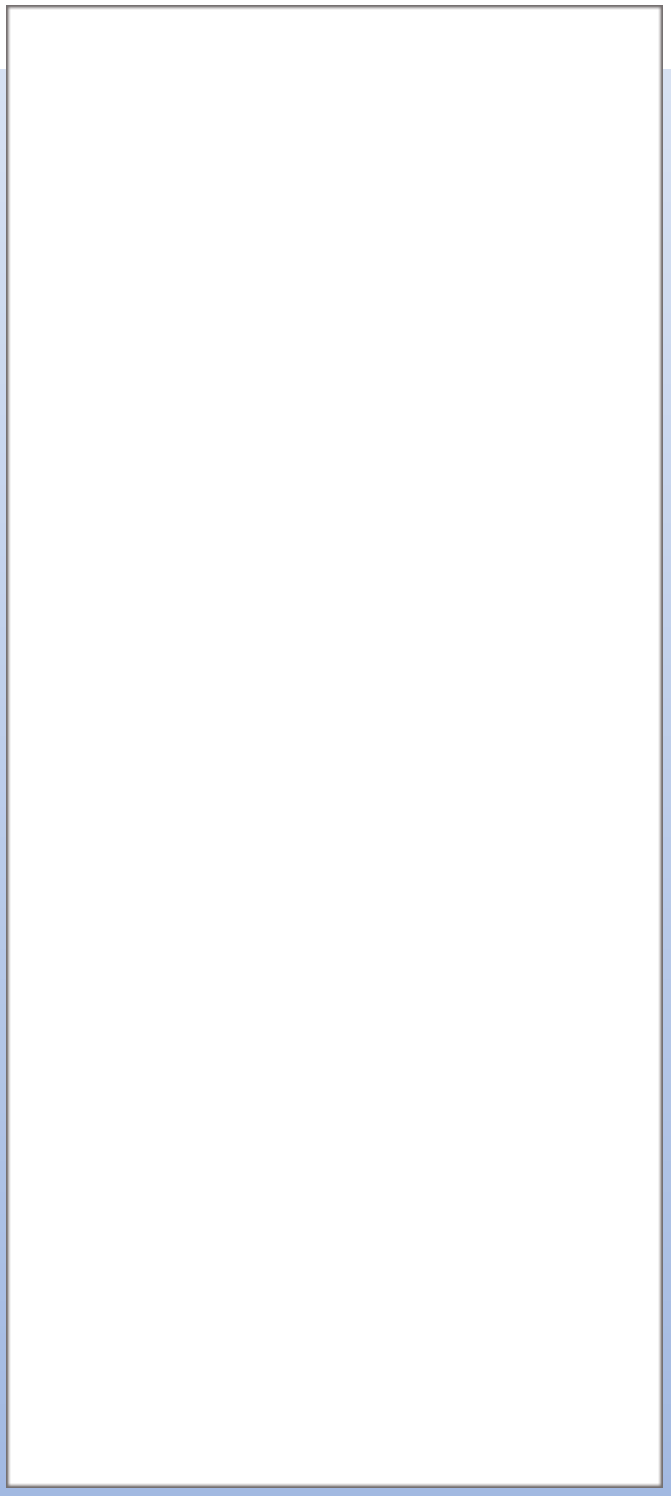


Design and develop simple graphics programs using basic graphics functions.

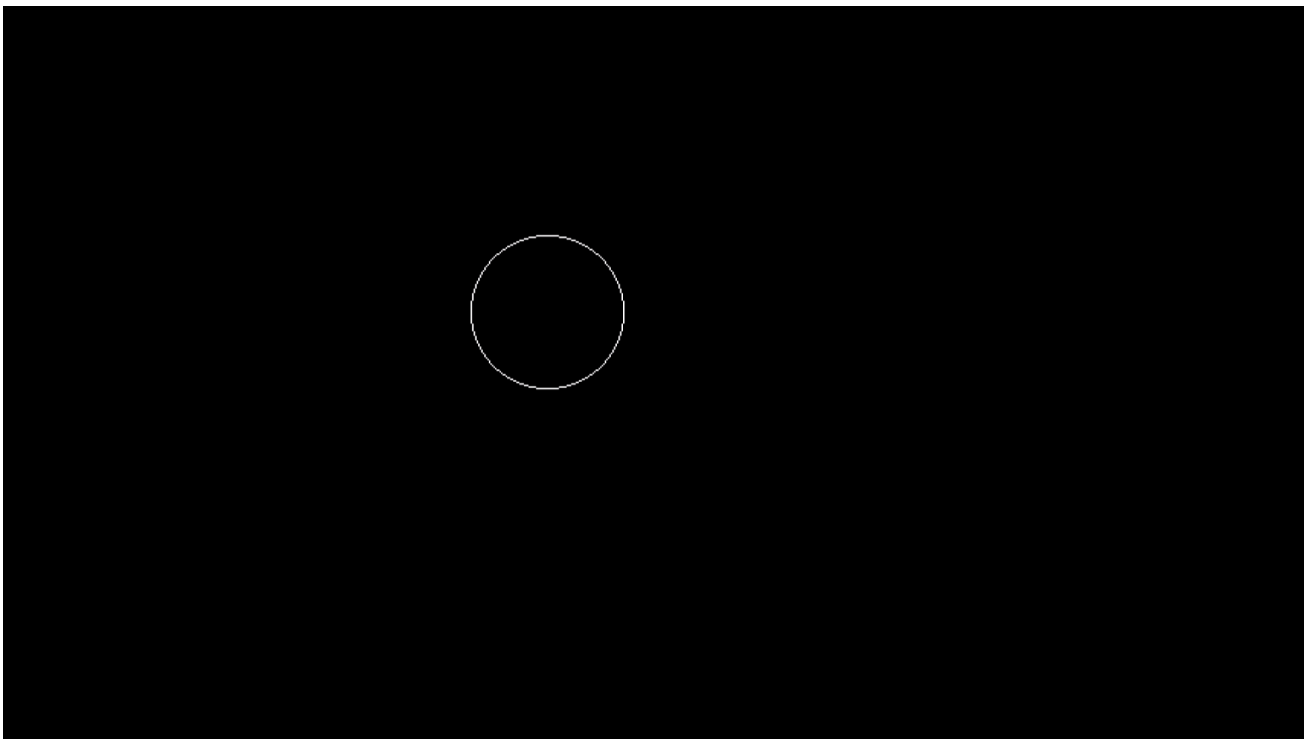
Draw line, circle, ellipse, rectangle, polygon using opengl graphics library. Use different colors for different shapes.

Circle:

Code:

```
#include <graphics.h>

//driver code
int main()
   {
   int gd = DETECT, gm;
     initgraph(&gd, &gm, "C:\\TURBOC3\\BGI");

   circle(250, 200, 50);

   getch();

   closegraph();

   return 0;
   }
```

OUTPUT:

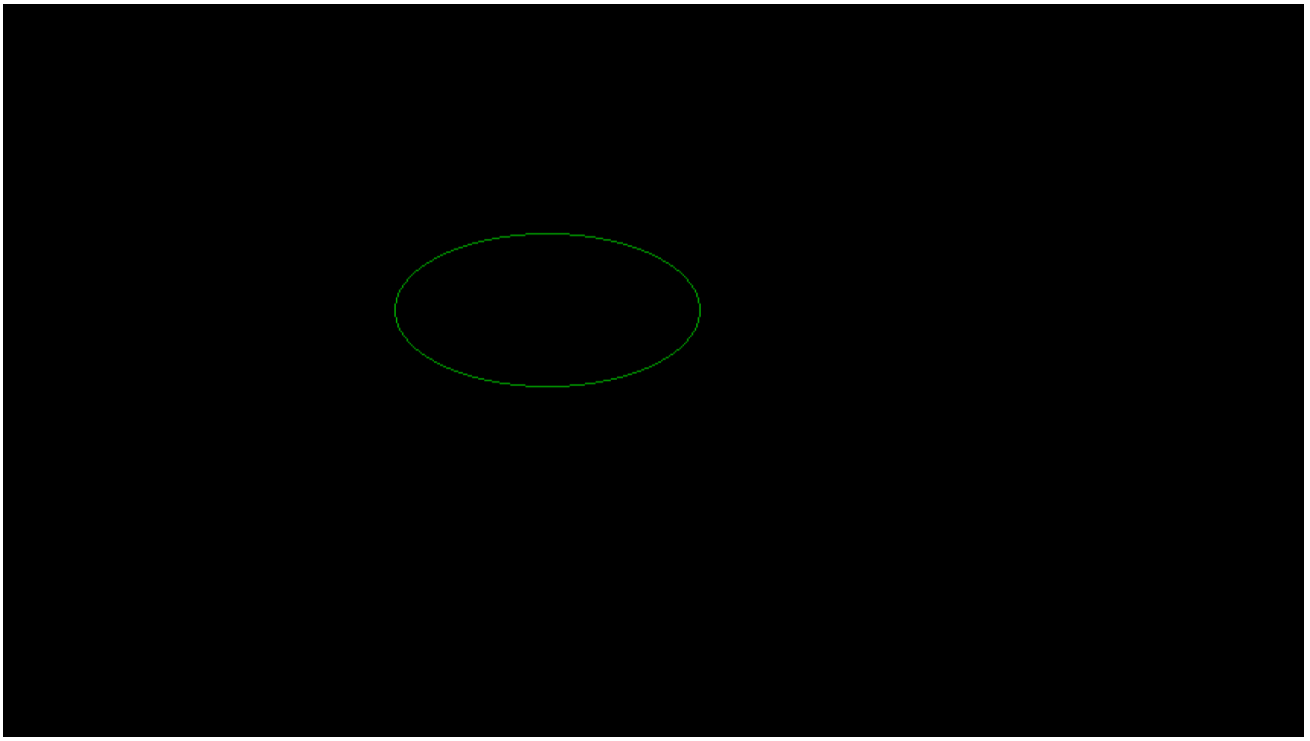


Ellipse :

Code:

```
#include <graphics.h>

int main()
{
        int gd = DETECT, gm;
        int x = 250, y = 200;
        int start_angle = 0;
        int end_angle = 360;
        int x_rad = 100;
        int y_rad = 50;
        initgraph(&gd, &gm, "C:\\TURBOC3\\BGI");
        setcolor(GREEN);
        ellipse(x, y, start_angle,
        end_angle, x_rad, y_rad);
        getch();
        closegraph();

        return 0;
}
```

OUTPUT:

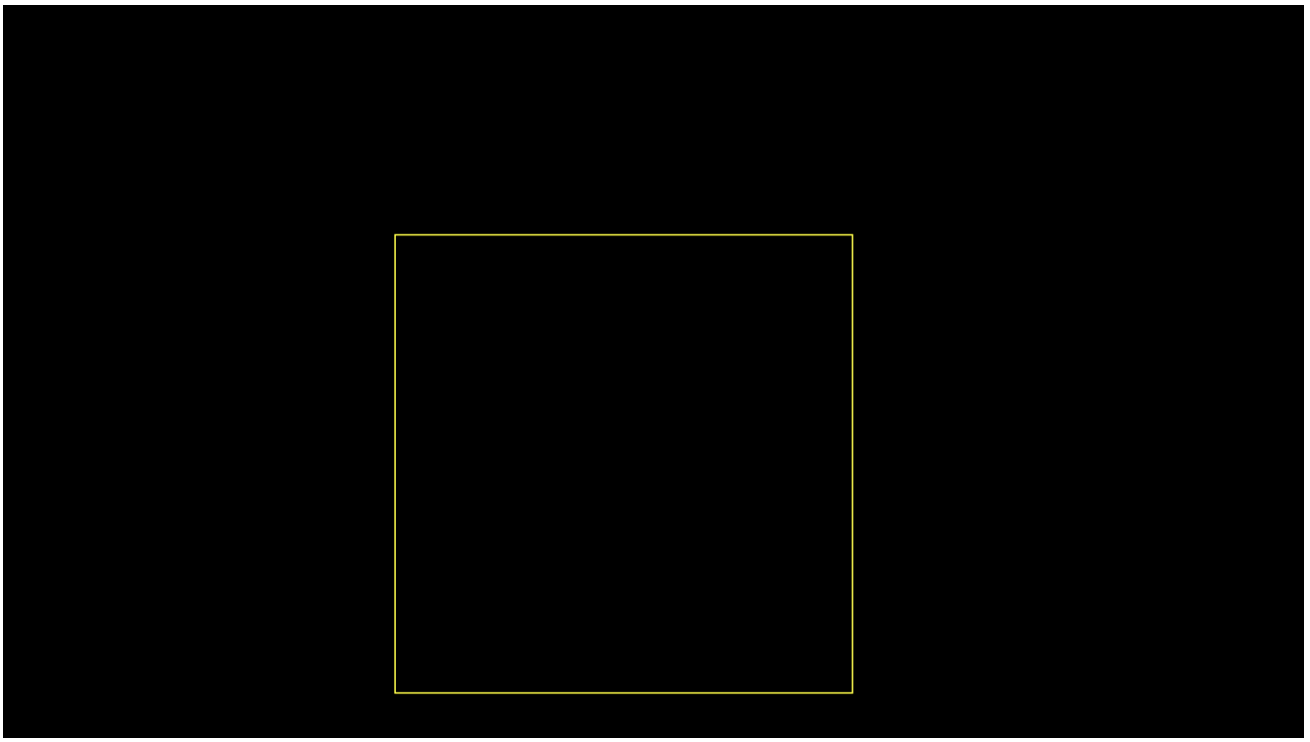


Rectangle:

Code:

```
// C program to draw a rectangle
#include <graphics.h>
int main()
{
   int gd = DETECT, gm;
   int left = 150, top = 150;
   int right = 450, bottom = 450;
   initgraph(&gd, &gm, "C:\\TURBOC3\\BGI");
   setcolor(YELLOW);
   rectangle(left, top, right, bottom);
   getch();
   closegraph();

   return 0;
}
```

OUTPUT:

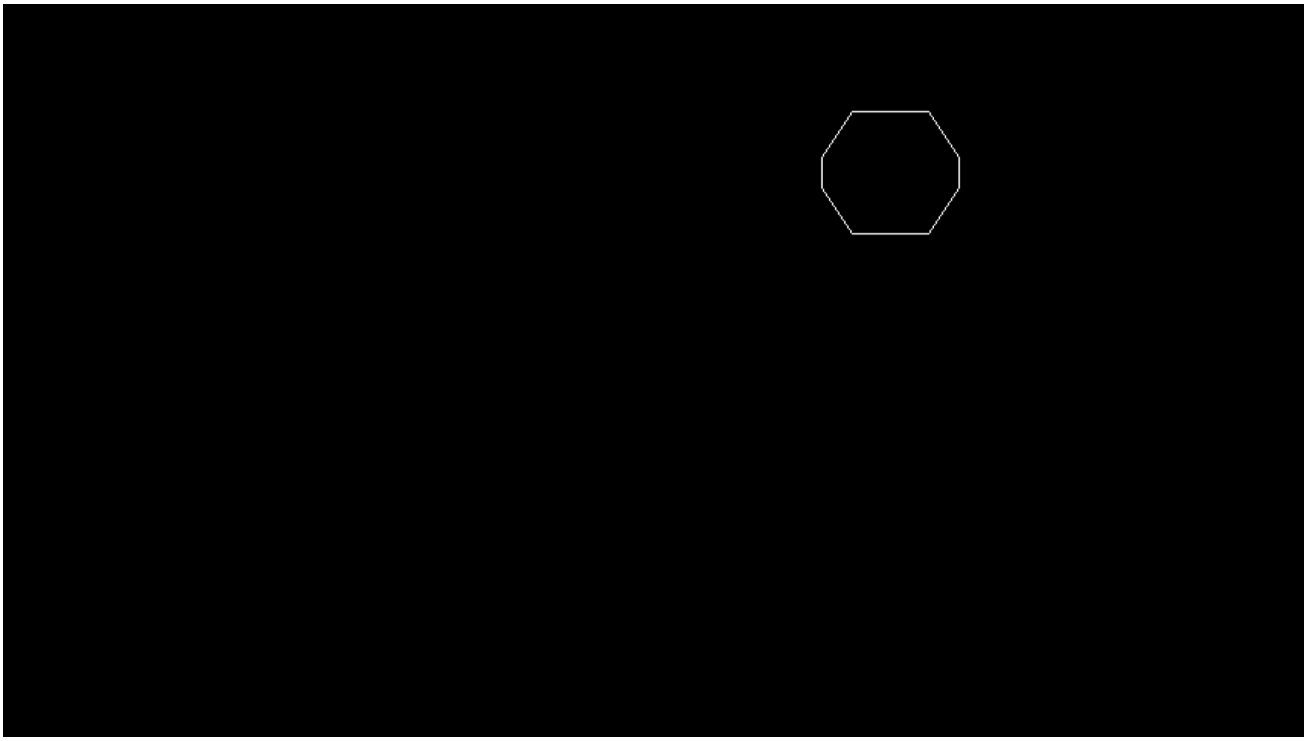


Polygon:

Code:

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

   int gd = DETECT, gm;
   int arr[] = {450,150, 430,120, 430,100, 450,70, 500,70, 520,100, 520,120, 500,150, 450,150};
   initgraph(&gd, &gm, "C:\\TURBOC3\\BGI");
   drawpoly(9, arr);

   getch();
   closegraph();

   return 0;
}
```

OUTPUT:



LAB 2

DEV SHAH(202000362)

Q1) draw a line using bresenham algorithm:

Code:

```
//Bresmen's Line Drawing algo
#include<stdio.h>
#include<graphics.h>
void drawline(int x0, int y0, int x1, int y1)
{
   int dx, dy, p, x, y;
   dx=x1-x0;
   dy=y1-y0;
   x=x0;
   y=y0;
   p=2*dy-dx;
   while(x<x1)
   {
      if(p>=0)
      {
         putpixel(x,y,7);
         y=y+1;
         p=p+2*dy-2*dx;
      }
      else
      {
         putpixel(x,y,7);
         p=p+2*dy;}
         x=x+1;
      }
}
int main()
{
   int gdriver=DETECT, gmode, error, x0, y0, x1, y1;
   initgraph(&gdriver, &gmode, "c:\\turboc3\\bgi");
   printf("Enter co-ordinates of first point: ");
   scanf("%d%d", &x0, &y0);
   printf("Enter co-ordinates of second point: ");
   scanf("%d%d", &x1, &y1);
   drawline(x0, y0, x1, y1);
   getch();
   return 0;

}
```

Output:

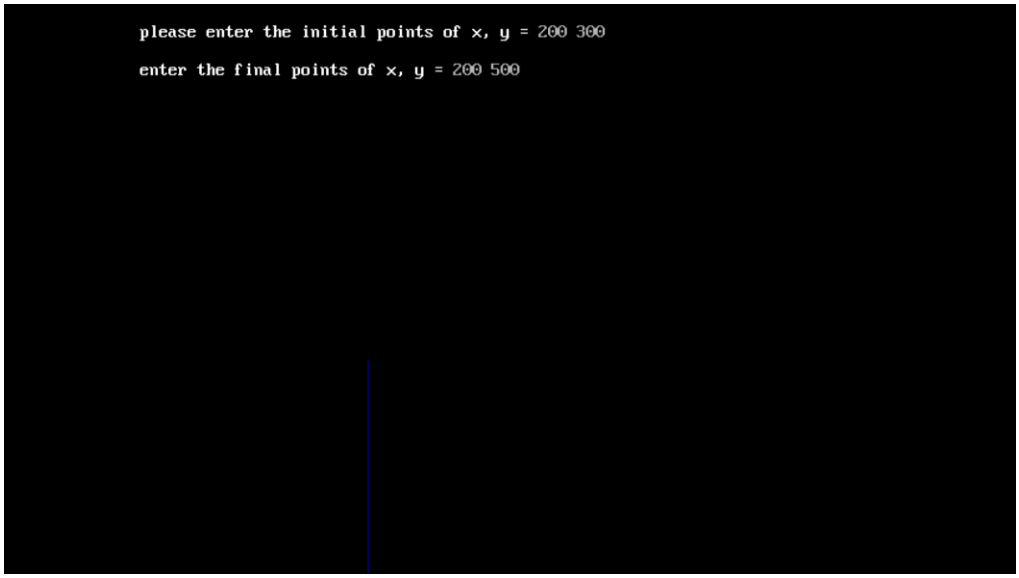

Enter co-ordinates of first point: 200 200
Enter co-ordinates of second point: 300 300

Q2) Draw a line using DDA algorithm.

Code:

```
#include<graphics.h>
#include<stdio.h>
#include<conio.h>
void main()
{
float x,y,x1,y1,delx,dely;
float slope;
int gr=DETECT,gm;

initgraph(&gr,&gm,"C:\\TURBOC3\\BGI");
printf("\n please enter the initial points of x, y = ");
scanf("%f %f",&x,&y);
printf("\n enter the final points of x, y = ");
scanf("%f %f",&x1,&y1);
dely= y1-y;
delx=x1-x;
slope=dely/delx;
if(slope>1.0)
{
while(y<=y1){
putpixel(x,y,1);
x=x+(1/slope);
y=y+1.0;
}
}

else{
while(x<=x1){
putpixel(x,y,1);
y=y+slope;
x=x+1.0;
}
}

getch();
}
```

Output:


please enter the initial points of x, y = 200 300
enter the final points of x, y = 200 500

Name: Dev Dinesh Shah

SRN: 202000362

Subject : Computer Graphics Lab

LAB 3

Implement Cohen Sutherland line clipping algorithm:

CODE:

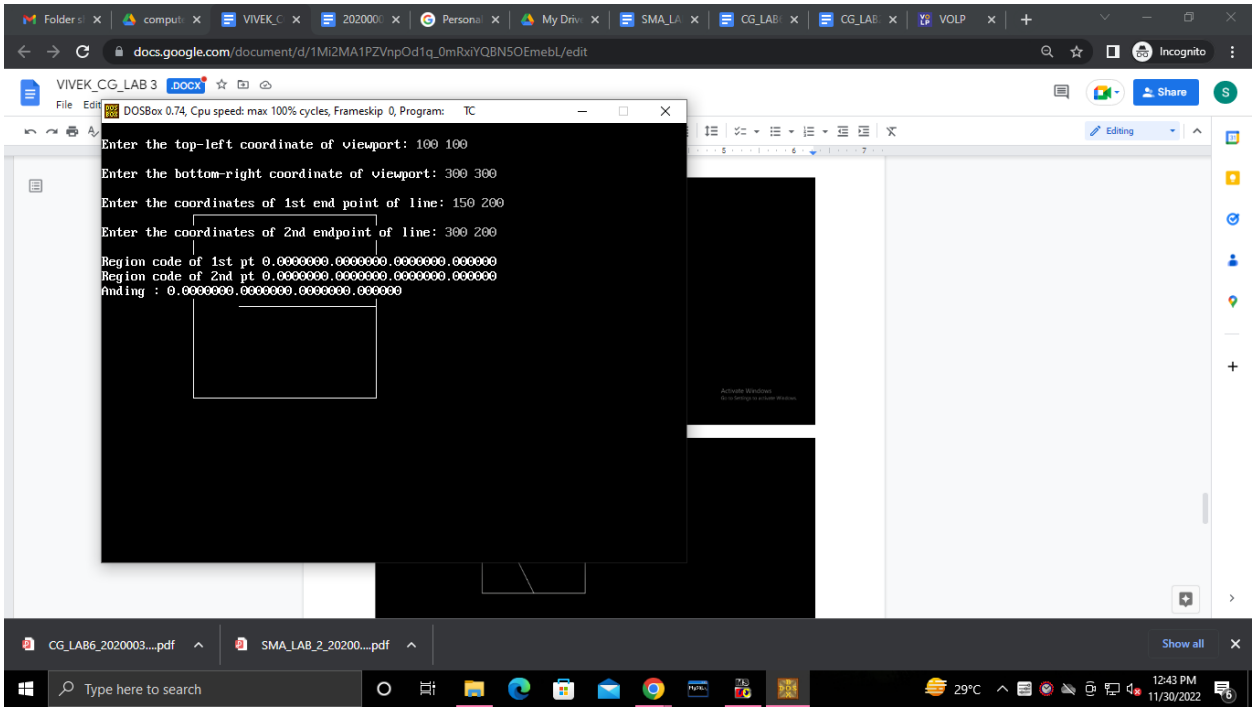
```
#include<stdio.h>
#include<graphics.h>
void main()
{
int gd=DETECT, gm;
float i,xmax,ymax,xmin,ymin,x11,y11,x22,y22,m;
float a[4],b[4],c[4],x1,y1;
clrscr();
initgraph(&gd,&gm,"c:\\turboc3\\bgi");
printf("\nEnter the top-left coordinate of viewport: ");
scanf("%f %f",&xmin,&ymin);
printf("\nEnter the bottom-right coordinate of viewport: ");
scanf("%f %f",&xmax,&ymax);
rectangle(xmin,ymin,xmax,ymax);
printf("\nEnter the coordinates of 1st end point of line: ");
scanf("%f %f",&x11,&y11);
printf("\nEnter the coordinates of 2nd endpoint of line: ");
scanf("%f %f",&x22,&y22);
line(x11,y11,x22,y22);
for(i=0;i<4;i++)
{
a[i]=0;
b[i]=0;
}
m=(y22-y11)/(x22-x11);
if(x11<xmin) a[3]=1;
```

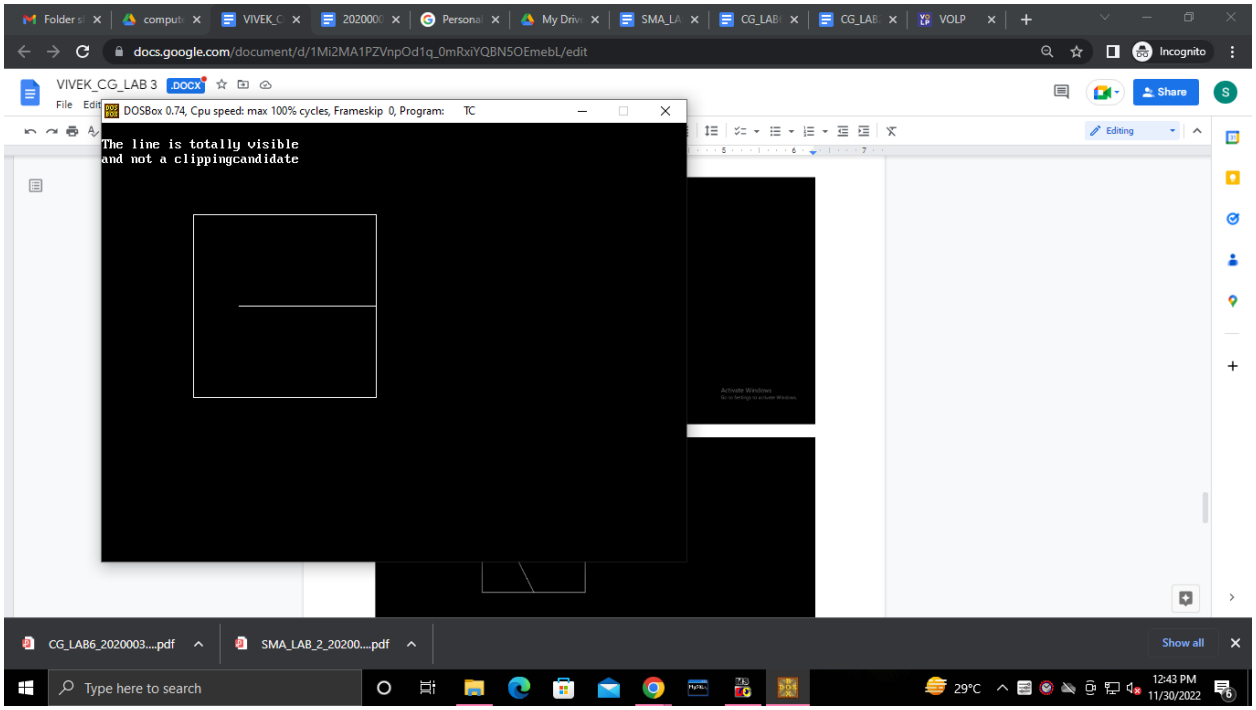
```
if(x11>xmax) a[2]=1;
if(y11<ymin) a[1]=1;
if(y11>ymax) a[0]=1;
if(x22<xmin) b[3]=1;
if(x22>xmax) b[2]=1;
if(y22<ymin) b[1]=1;
if(y22>ymax) b[0]=1;
printf("\nRegion code of 1st pt ");
for(i=0;i<4;i++)
{
printf("%f",a[i]);
}
printf("\nRegion code of 2nd pt ");
for(i=0;i<4;i++)
{
printf("%f",b[i]);
}
printf("\nAnding : ");
for(i=0;i<4;i++)
{
c[i]=a[i]&&b[i];
}
for(i=0;i<4;i++)
printf("%f",c[i]);
getch();
if((c[0]==0)&&(c[1]==0)&&(c[2]==0)&&(c[3]==0))
{
if((a[0]==0)&&(a[1]==0)&&(a[2]==0)&&(a[3]==0)&&(b[0]==0)&&
(b[1]==0)&&(b[2]==0)&&(b[3]==0))
{
clrscr();
clearviewport();
printf("\nThe line is totally visible\nand not a clippingcandidate");
rectangle(xmin,ymin,xmax,ymax);
line(x11,y11,x22,y22);
getch();
}
else
{
clrscr();
clearviewport();
printf("\nLine is partially visible");
rectangle(xmin,ymin,xmax,ymax);
line(x11,y11,x22,y22);
```

```
getch();
if((a[0]==0)&&(a[1]==1))
{
x1=x11+(ymin-y11)/m;
x11=x1;
y11=ymin;
}
else if((b[0]==0)&&(b[1]==1))
{
x1=x22+(ymin-y22)/m;
x22=x1;
y22=ymin;
}
if((a[0]==1)&&(a[1]==0))
{
x1=x11+(ymax-y11)/m;
x11=x1;
y11=ymax;
}
else if((b[0]==1)&&(b[1]==0))
{
x1=x22+(ymax-y22)/m;
x22=x1;
y22=ymax;
}
if((a[2]==0)&&(a[3]==1))
{
y1=y11+m*(xmin-x11);
y11=y1;
x11=xmin;
}
else if((b[2]==0)&&(b[3]==1))
{
y1=y22+m*(xmin-x22);
y22=y1;
x22=xmin;
}
if((a[2]==1)&&(a[3]==0))
{
y1=y11+m*(xmax-x11);
y11=y1;
x11=xmax;
}
else if((b[2]==1)&&(b[3]==0))
```

```
{
y1=y22+m*(xmax-x22);
y22=y1;
x22=xmax;
}
clrscr();
clearviewport();
printf("\nAfter clippling:");
rectangle(xmin,ymin,xmax,ymax);
line(x11,y11,x22,y22);
getch();
}
}
else
{
clrscr();
clearviewport();
printf("\nLine is invisible");
rectangle(xmin,ymin,xmax,ymax);
getch();
}
closegraph();
getch();
}
```

Output:




docs.google.com/document/d/1Mi2MA1PZVnpOd1q_0mRxiYQBN5OEmebL/edit
VIVEK_CG_LAB 3
DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC
The line is totally visible
and not a clippingcandidate
CG_LAB6_2020003....pdf
SMA_LAB_2_20200....pdf
Show all
Type here to search
29°C
12:43 PM
11/30/2022

Assignment 4

Dev Shah

202000362

Network Security Lab

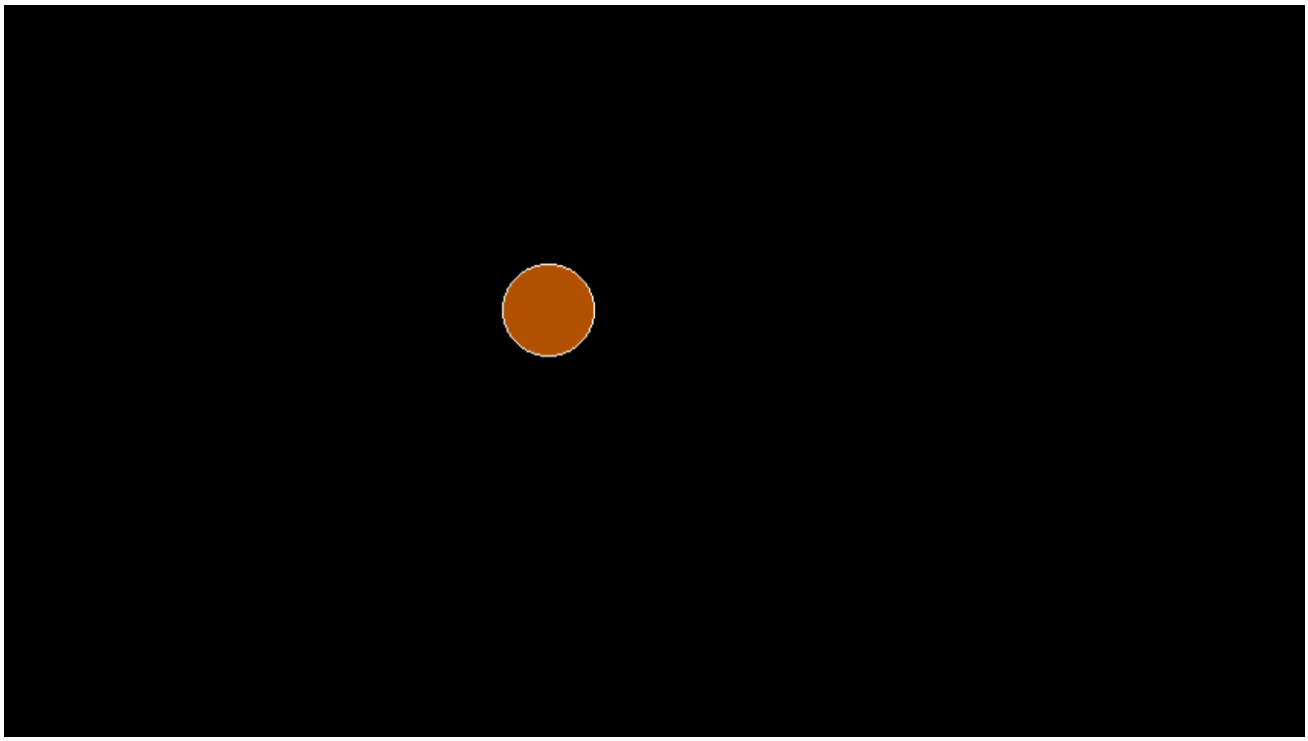
4 connect Algorithm

Code:

```
// C Implementation for Boundary Filling Algorithm
#include <graphics.h>

// Function for 4 connected Pixels
void boundaryFill4(int x, int y, int fill_color,int boundary_color)
{
   if(getpixel(x, y) != boundary_color &&
     getpixel(x, y) != fill_color)
   {
      putpixel(x, y, fill_color);
      boundaryFill4(x + 1, y, fill_color, boundary_color);
      boundaryFill4(x, y + 1, fill_color, boundary_color);
      boundaryFill4(x - 1, y, fill_color, boundary_color);
           boundaryFill4(x, y - 1, fill_color, boundary_color);

   }
}

//driver code
int main()
{
   // gm is Graphics mode which is
   // a computer display mode that
   // generates image using pixels.
   // DETECT is a macro defined in
   // "graphics.h" header file
   int gd = DETECT, gm;

   // initgraph initializes the
   // graphics system by loading a
   // graphics driver from disk
   initgraph(&gd, &gm, "c:\\turboc3\\bgi");

   //int x = 250, y = 200, radius = 50;

   // circle function
   circle(250, 200, 30);

   // Function calling
   boundaryFill4(250, 200, 6, 15);

   delay(10000);

   getch();

   // closegraph function closes the
   // graphics mode and deallocates
   // all memory allocated by
   // graphics system .
   closegraph();
```

```
    return 0;
}
```

Output:



8 Connect Algorithm:

Code:

```
// C Implementation for Boundary Filling Algorithm
#include <graphics.h>

// Function for 8 connected Pixels
void boundaryFill8(int x, int y, int fill_color,int boundary_color)
{
	if(getpixel(x, y) != boundary_color &&
	getpixel(x, y) != fill_color)
	{
		putpixel(x, y, fill_color);
		boundaryFill8(x + 1, y, fill_color, boundary_color);
		boundaryFill8(x, y + 1, fill_color, boundary_color);
		boundaryFill8(x - 1, y, fill_color, boundary_color);
		boundaryFill8(x, y - 1, fill_color, boundary_color);
		boundaryFill8(x - 1, y - 1, fill_color, boundary_color);
		boundaryFill8(x - 1, y + 1, fill_color, boundary_color);
		boundaryFill8(x + 1, y - 1, fill_color, boundary_color);
		boundaryFill8(x + 1, y + 1, fill_color, boundary_color);
	}
}

//driver code
int main()
{
	// gm is Graphics mode which is
	// a computer display mode that
	// generates image using pixels.
```

```
        // DETECT is a macro defined in
        // "graphics.h" header file
        int gd = DETECT, gm;

        // initgraph initializes the
        // graphics system by loading a
        // graphics driver from disk
        initgraph(&gd, &gm, "c:\\turboc3\\bgi");

        // Rectangle function
        rectangle(50, 50, 100, 100);

        // Function calling
        boundaryFill8(55, 55, 4, 15);

        delay(10000);

        getch();

        // closegraph function closes the
        // graphics mode and deallocates
        // all memory allocated by
        // graphics system .
        closegraph();

        return 0;
}
```

OUTPUT:

Floodfill algorithm

Code:

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

```
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>
#include <math.h>

void floodFill(int x,int y,int oldcolor,int newcolor)
{
if(getpixel(x,y) == oldcolor)
{
delay(1);
putpixel(x,y,newcolor);
floodFill(x,y-1,oldcolor,newcolor);
floodFill(x,y+1,oldcolor,newcolor);
floodFill(x-1,y,oldcolor,newcolor);
floodFill(x+1,y,oldcolor,newcolor);

}
}

void EightWaySymmetricPlot(int xc,int yc,int x,int y)
{
   putpixel(x+xc,y+yc,CYAN);
   putpixel(x+xc,-y+yc,RED);
   putpixel(-x+xc,-y+yc,YELLOW);
   putpixel(-x+xc,y+yc,GREEN);
   putpixel(y+xc,x+yc,14);
   putpixel(y+xc,-x+yc,12);
   putpixel(-y+xc,-x+yc,13);
   putpixel(-y+xc,x+yc,7);
}

void BresenhamCircle(int xc,int yc,int r)
{
   int x=0,y=r,d=3-(2*r);
   EightWaySymmetricPlot(xc,yc,x,y);

   while(x<=y)
    {
     if(d<=0)
          {
       d=d+(4*x)+6;
     }
    else
     {
       d=d+(4*x)-(4*y)+10;
       y=y-1;
     }
      x=x+1;
      EightWaySymmetricPlot(xc,yc,x,y);
     }
   }

   int  main(void)
  {
```

```
int xc,yc,r,gdriver = DETECT, gmode;
/* initialize graphics and local variables */
 initgraph(&gdriver, &gmode, "C:\\TURBOC3\\BGI");

  printf("Enter the values of xc and yc :");
  scanf("%d%d",&xc,&yc);
  printf("Enter the value of radius  :");
  scanf("%d",&r);
  BresenhamCircle(xc,yc,r);
        floodFill(xc-r+2,yc-2,0,YELLOW);//0=black, 1=blue
        delay(500);
 getch();
 closegraph();
 return 0;
}
```

OUTPUT:

Assignment 5

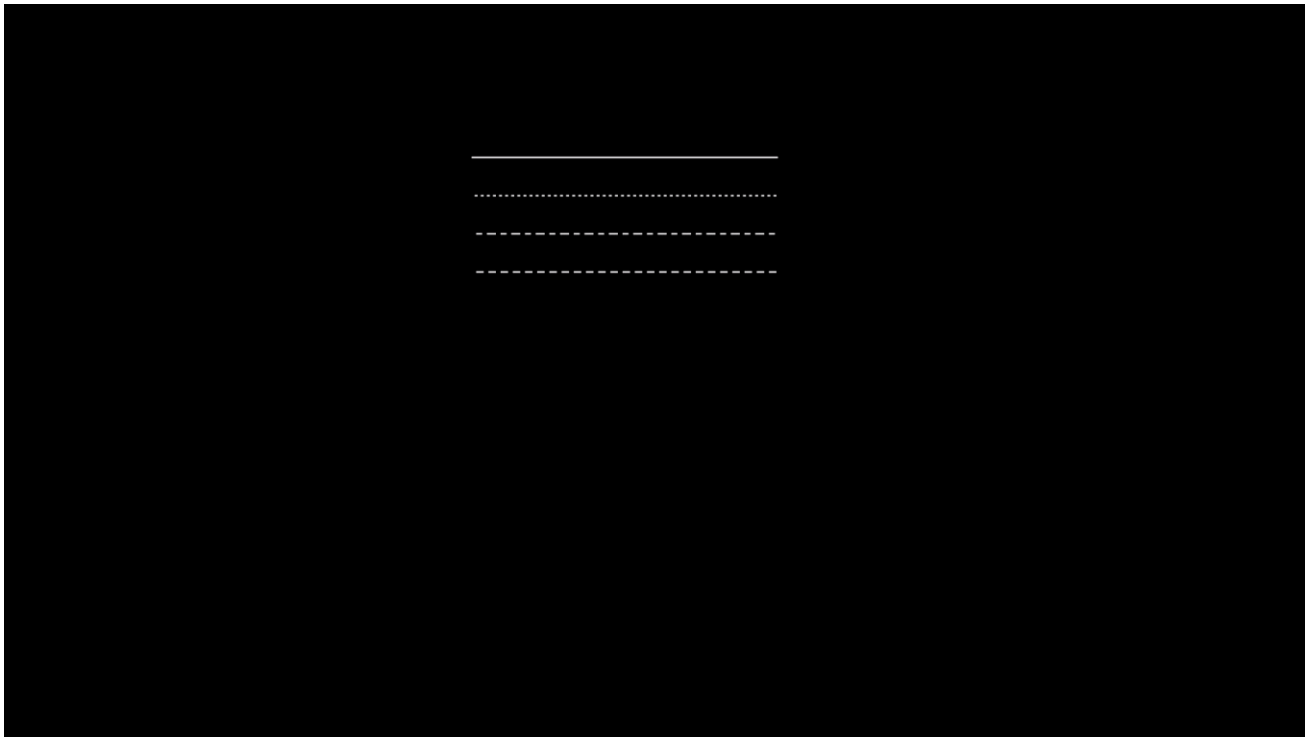
DEV SHAH(202000362)

Q.1 Draw different styled lines using computer graphics

Code:

```
#include <graphics.h>
// driver code
int main()
{
// gm is Graphics mode which is
// a computer display mode that
// generates image using pixels.
// DETECT is a macro defined in
// "graphics.h" header file
int gd = DETECT, gm;
// variable to change the
// line styles
int c;
// initial coordinate to
// draw line
int x = 200, y = 100;
// initgraph initializes the
// graphics system by loading a
// graphics driver from disk
initgraph(&gd, &gm, "C:\\TURBOC3\\BGI" );
// To keep track of lines
for ( c = 0 ; c < 5 ; c++ )
{
// setlinestyle function
setlinestyle(c, 0, 1);
// Drawing line
line(x, y, x+200, y);
y = y + 25;
}
getch();
// closegraph function closes the
// graphics mode and deallocates
// all memory allocated by
// graphics system .
closegraph();
return 0;
}
```

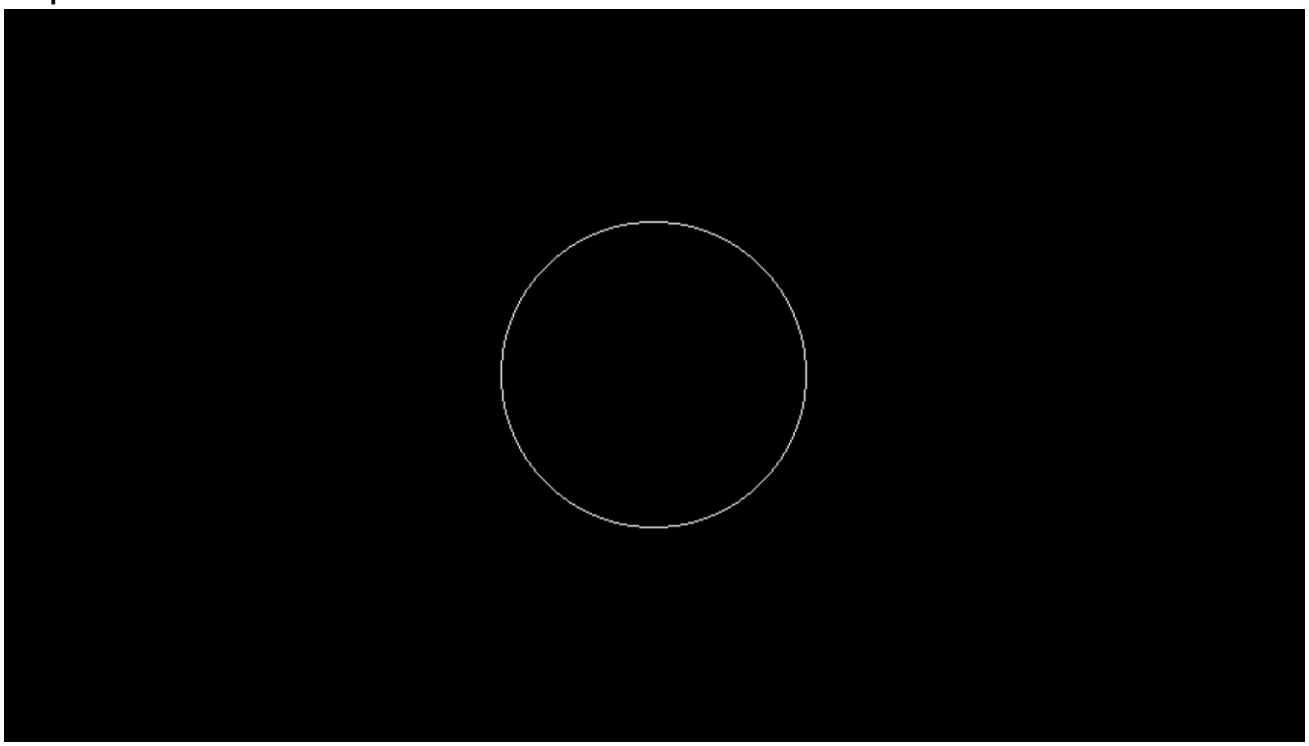
Output:



Q.2 Draw circle at center of screen.

```
#include<stdio.h>
#include<graphics.h>
#include<conio.h>
int main(){
 int gd = DETECT,gm;
 int x ,y ,radius=100;
 clrscr();
 initgraph(&gd, &gm, "C:\\TURBOC3\\BGI");
 /* Initialize center of circle with center of screen */
 x = getmaxx()/2;
 y = getmaxy()/2;
 //outtextxy(x-100, 50, "circle at center");
 /* Draw circle on screen */
 circle(x, y, radius);
 getch();
 closegraph();
 return 0;
}
```

Output:

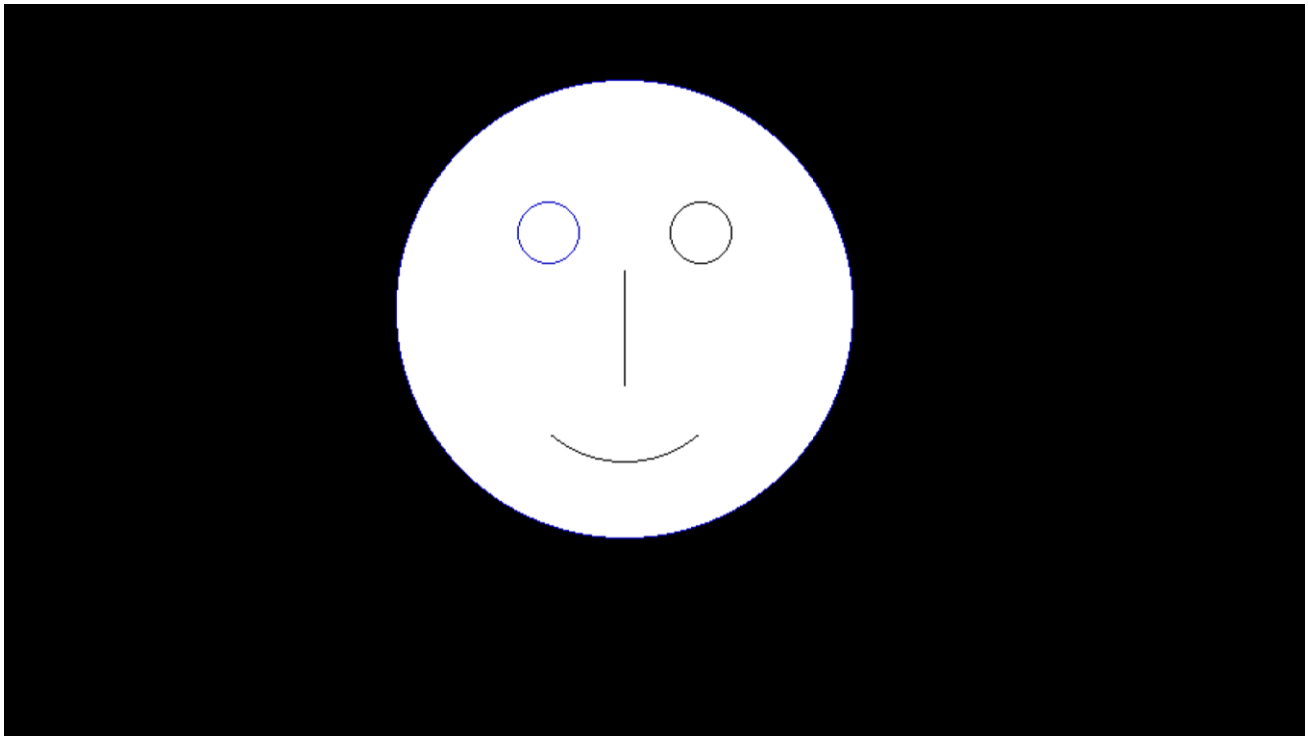


Q.3 Draw human face(smiley) using computer graphics.

Code:

```
#include<stdio.h>
#include<graphics.h>
#include<conio.h>
#include<dos.h>
int main(){
int gd = DETECT, gm;
initgraph(&gd, &gm, (char*) "C:\\TURBOC3\\BGI" );
setcolor(BLUE);
circle(300,200,150);
//setfillstyle(SOLID_FILL, BLUE);
floodfill(300, 100, BLUE);
circle(250,150,20);
setcolor(BLACK);
setfillstyle(SOLID_FILL, BLACK);
circle(350,150,20);
setcolor(BLACK);
setfillstyle(SOLID_FILL, BLACK);
line(300,175,300,250);
arc(300,225,230,310,75);
getch();
closegraph();
return 0;
}
```

Output:



Q.4 Display random circles on screen using computer graphics.

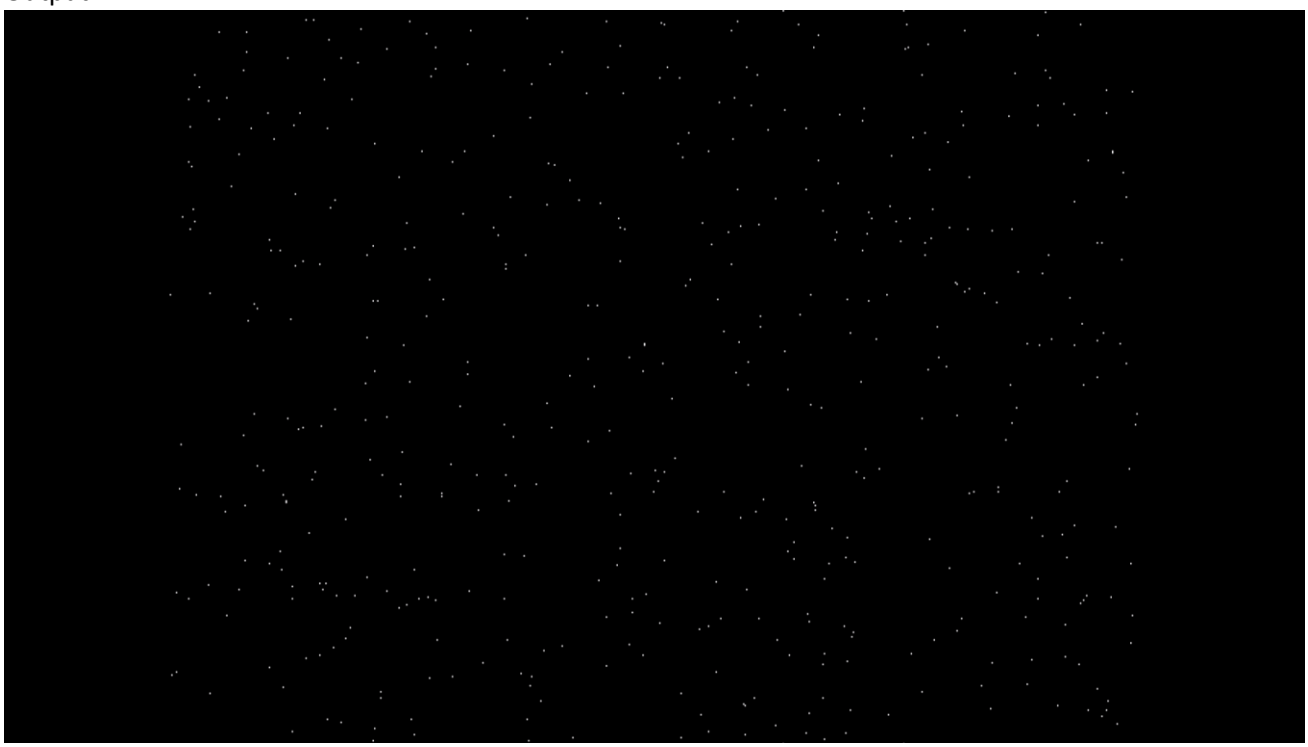
Code:

Q.5 Display random pixels on screen using computer graphics.

Code:

```
#include <conio.h>
#include <graphics.h>
#include <dos.h>
#include <stdlib.h>
int main() {
 int gd = DETECT, gm;
 int i, x, y;
 initgraph(&gd, &gm, "C:\\TURBOC3\\BGI");

while (!kbhit()) {
 /* color 500 random pixels on screen */
 for(i=0; i<=500; i++) {
 x=rand()%getmaxx();
 y=rand()%getmaxy();
 putpixel(x,y,15);
 }
 delay(500);
 /* clears screen */
 cleardevice();
 }
 getch();
 closegraph();
 return 0;
}
```

Output:



Name: Dev DInesh Shah

SRN: 202000362

Subject : Computer Graphics

LAB 2

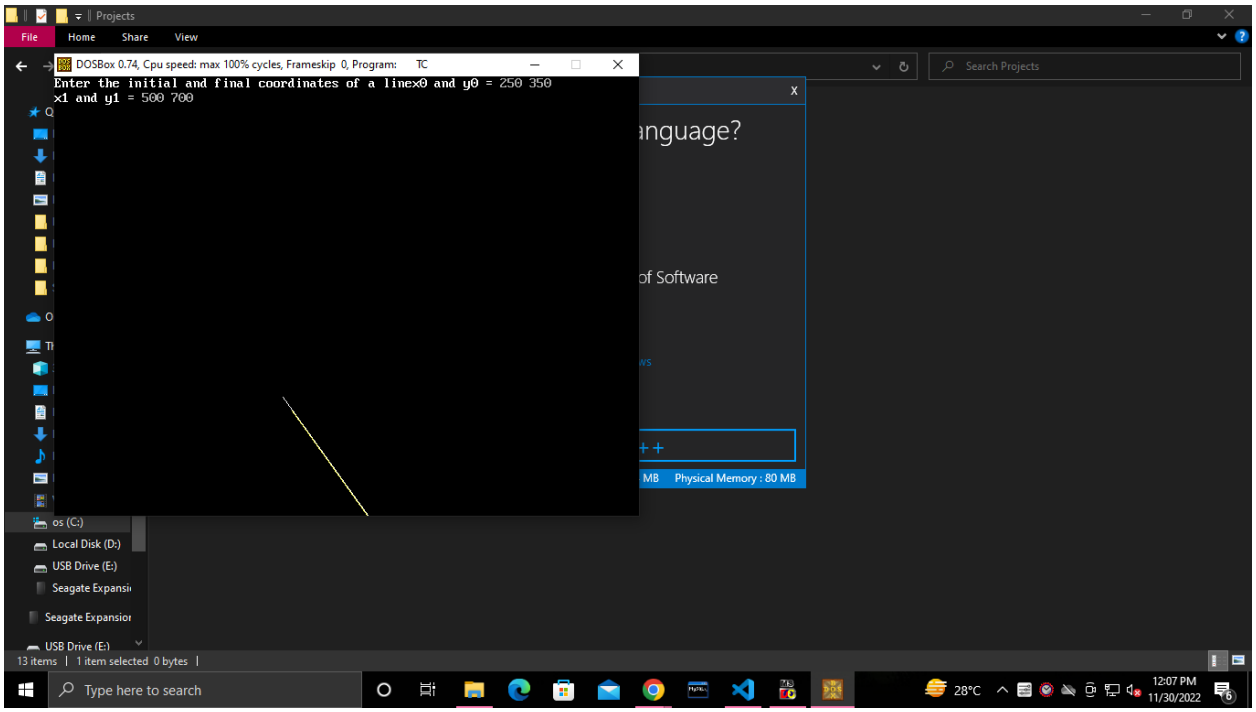
1.TRANSLATION Of A LINE IN C

Source code:

```
#include<conio.h>
#include<graphics.h>
#include<stdio.h>
void main()
{
int gd=DETECT,gm;
// declaring two array
// Translation vectoralready initialized
int l[2][2],v[2]={10,15},i=0,j;
clrscr();
initgraph(&gd,&gm,"C:\\TURBOC3\\BGI");
printf("Enter theinitialand finalcoordinates of a
line");
// Getting input from user, having 2D array
where1st row representsinitial point
// And Second row representsfinalcoordinate
while(i<2)
```

```
{
printf("x%d and y%d = ",i,i);
j=0;
scanf("%d",&l[i][j]);
scanf("%d",&l[i][j+1]);
i++;
}
// Line beforetranslation
line(l[0][0],l[0][1],l[1][0],l[1][1]);
setcolor(YELLOW);
// Line after translation
line(l[0][0]+v[0],l[0][1]+v[1],l[1][0]+v[0],l[1][1]
+v[1]); // Adding Translation vector in it to
changethe position
getch();
closegraph();
}
```

Output screen:



2. SCALING Of AN OBJECT IN C .

SOURCE CODE:

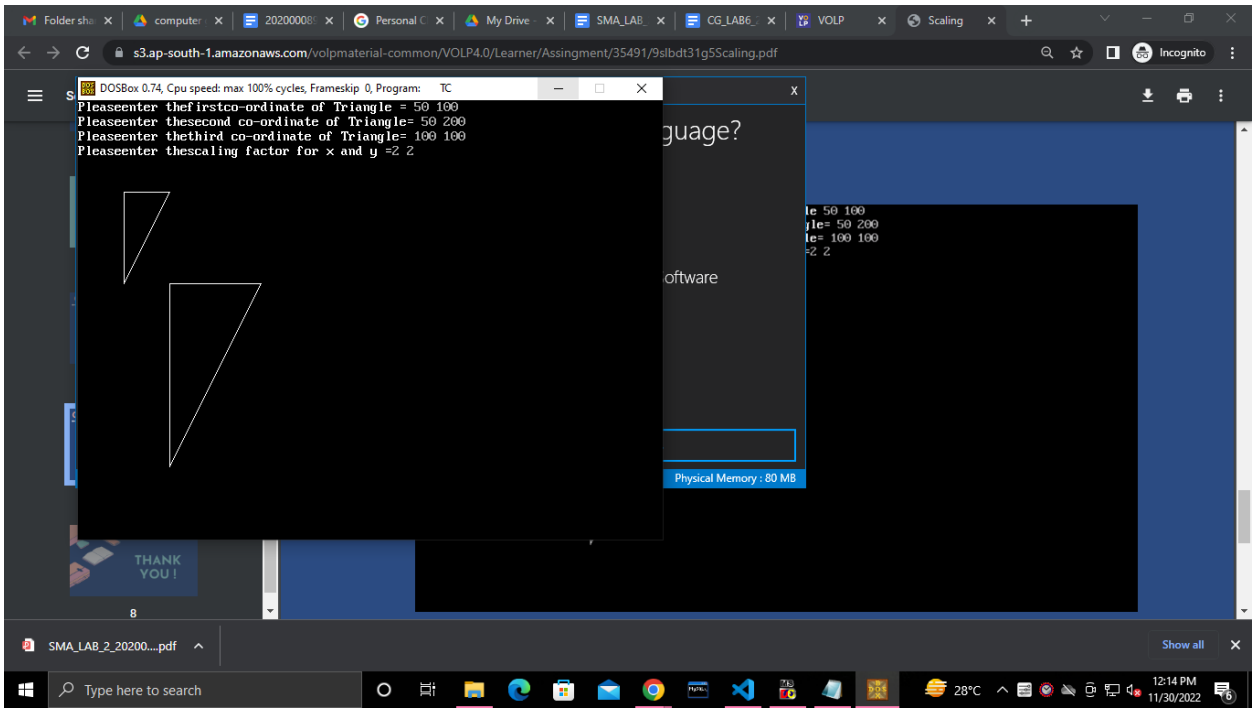
```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
void main()
{
```

```
int x,y,x1,y1,x2,y2;
int s_x,s_y;
int gd=DETECT,gm;
initgraph(&gd,&gm,"c:\\TURBOC3\\BGI" );
printf("Pleaseenter thefirstco-ordinate of Triangle = ");
scanf("%d %d",&x,&y);
printf("Pleaseenter thesecond co-ordinate of Triangle= ");
scanf("%d %d",&x1,&y1);
printf("Pleaseenter thethird co-ordinate of Triangle= ");
scanf("%d %d",&x2,&y2);
line(x,y,x1,y1);
```

```
line(x1,y1,x2,y2);
line(x2,y2,x,y);
printf("Pleaseenter thescaling factor for
x and y =");
scanf("%d %d", &s_x,&s_y);
x=x*s_x;
x1=x1*s_x;
x2=x2*s_x;
y=y*s_y;
y1=y1*s_y;
y2=y2*s_y;
line(x,y,x1,y1);
line(x1,y1,x2,y2);
line(x2,y2,x,y);
getch();
```

```
closegraph();
}
```

Output screen:



3. ROTATION Of AN OBJECT IN C.

SOURCE CODE:

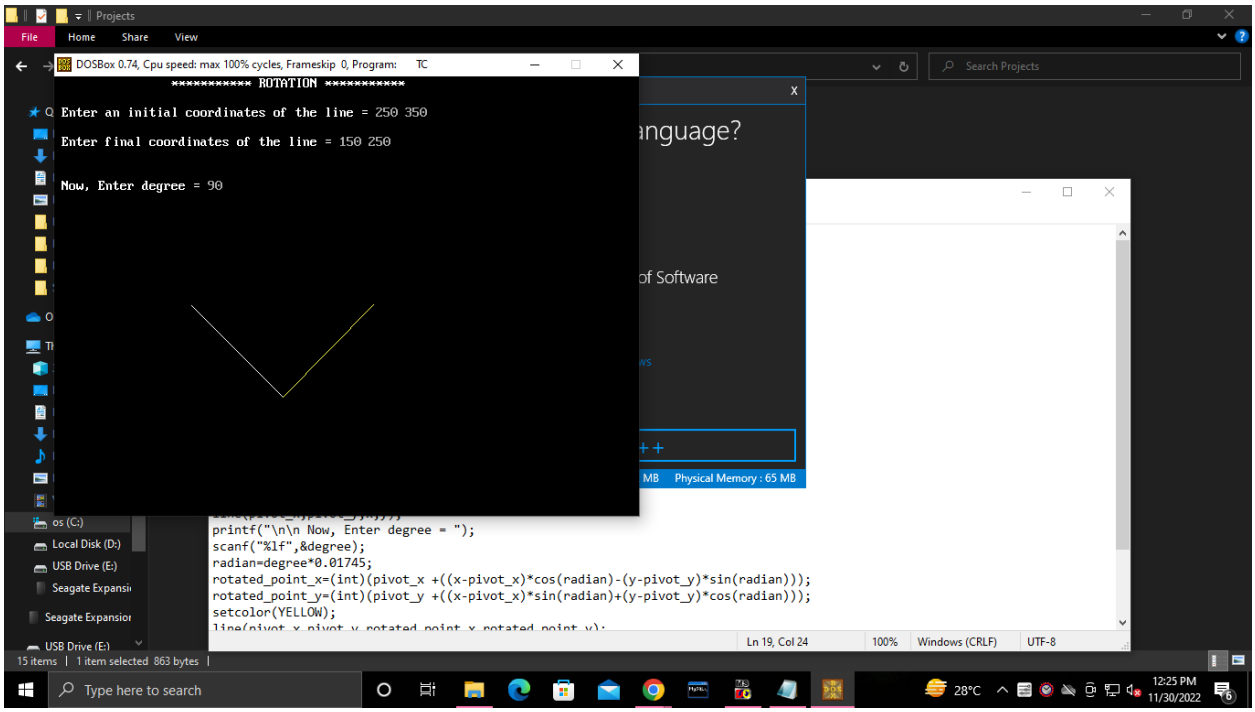
```
#include<graphics.h>
#include<stdio.h>
```

```
#include<conio.h>
#include<math.h>
void main()
{
int gd=DETECT,gm;
int pivot_x,pivot_y,x,y;
double degree,radian;
int rotated_point_x,rotated_point_y;
initgraph(&gd,&gm,"C://TURBOC3/
/BGI");
cleardevice();
printf("\t\t********** ROTATION
********** \n");
printf("\n Enteran initialcoordinates
of theline = ");
```

```
scanf("%d %d",&pivot_x,&pivot_y);
printf("\n Enter the  final coordinates of the line = ");
scanf("%d %d",&x,&y);
line(pivot_x,pivot_y,x,y);
printf("\n\n Now, Enter  the degree = ");
scanf("%lf",&degree);
radian=degree*0.01745;
rotated_point_x=(int)(pivot_x +((x-pivot_x)*cos(radian)-(y-pivot_y )*sin(radian)));
rotated_point_y=(int)(pivot_y +((x-pivot_x)*sin(radian)+(y-pivot_ y)*cos(radian)));
setcolor(YELLOW);
```

```
line(pivot_x,pivot_y,rotated_point_x
,rotated_point_y);
getch();
closegraph();
}
```

Output screen:



4. REFLECTION Of AN OBJECT IN C .

SOURCE CODE:

```
// C program for the aboveapproach
#include<conio.h>
#include<graphics.h>
#include<stdio.h>
// Driver Code
void main()
{
// Initializethe drivers
int gm, gd = DETECT, ax, x1 = 100;
int x2 = 100, x3 = 200, y1 = 100;
int y2 = 200, y3 = 100;
// Add in your BGI folder path
```

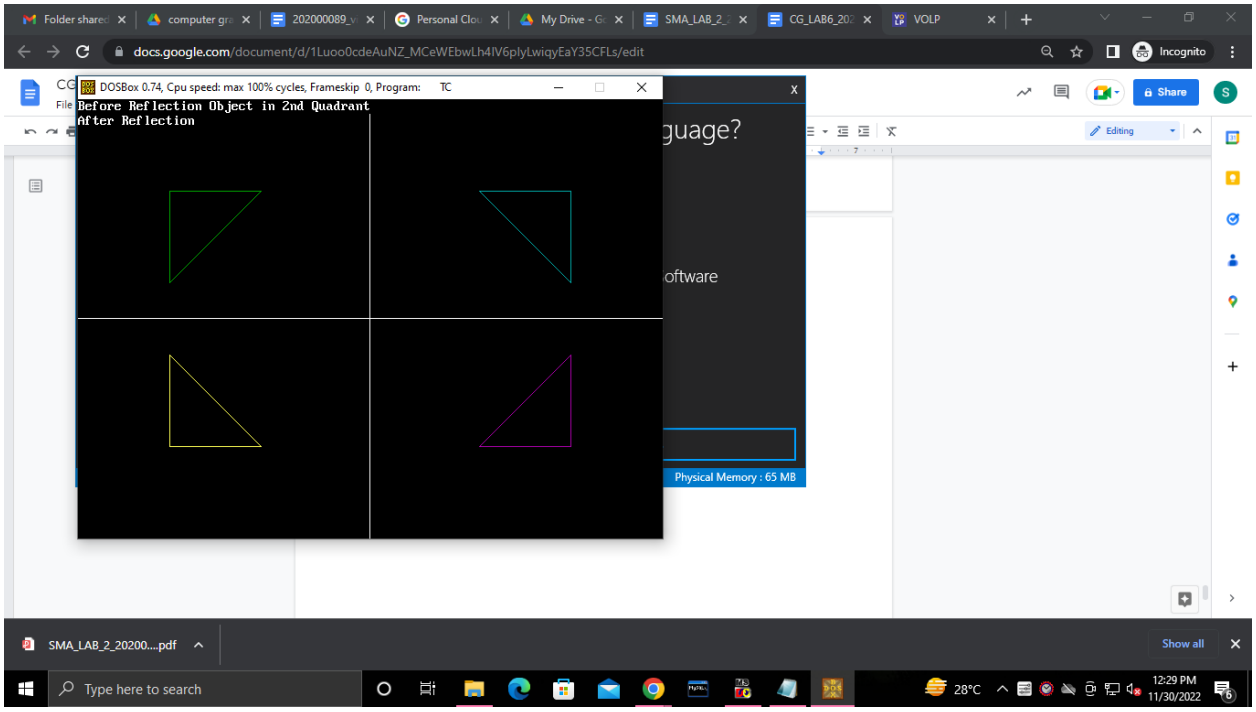
```
initgraph(&gd, &gm,
"C:\\TURBOC3\\BGI");
cleardevice();
// Draw the graph
line(getmaxx() / 2, 0, getmaxx() / 2,
getmaxy());
line(0, getmaxy() / 2, getmaxx(),
getmaxy() / 2);
// Object initially at 2nd quadrant
printf("Before Reflection Object"
" in 2nd Quadrant");
// Set thecolor
setcolor(14);
line(x1, y1, x2, y2);
line(x2, y2, x3, y3);
line(x3, y3, x1, y1);
```

```
getch();
// After reflection
printf("\nAfter Reflection");
// Reflection along origin i.e., // in 4th quadrant
setcolor(4);
line(getmaxx() - x1, getmaxy() - y1, getmaxx() - x2, getmaxy() - y2);
line(getmaxx() - x2, getmaxy() - y2, getmaxx() - x3, getmaxy() - y3);
line(getmaxx() - x3, getmaxy() - y3, getmaxx() - x1, getmaxy() - y1);
// Reflection along x-axis i.e., // in 1st quadrant
setcolor(3);
line(getmaxx() - x1, y1, getmaxx() - x2, y2);
```

```
line(getmaxx() - x2, y2, getmaxx() -
x3, y3);
line(getmaxx() - x3, y3, getmaxx() -
x1, y1);
// Reflection along y-axis i.e., // in
3rd quadrant
setcolor(2);
line(x1, getmaxy() - y1, x2, getmaxy()
- y2);
line(x2, getmaxy() - y2, x3, getmaxy()
- y3);
line(x3, getmaxy() - y3, x1,
getmaxy() - y1);
getch();
// Closethe graphics
closegraph();
```

}

Output screen:



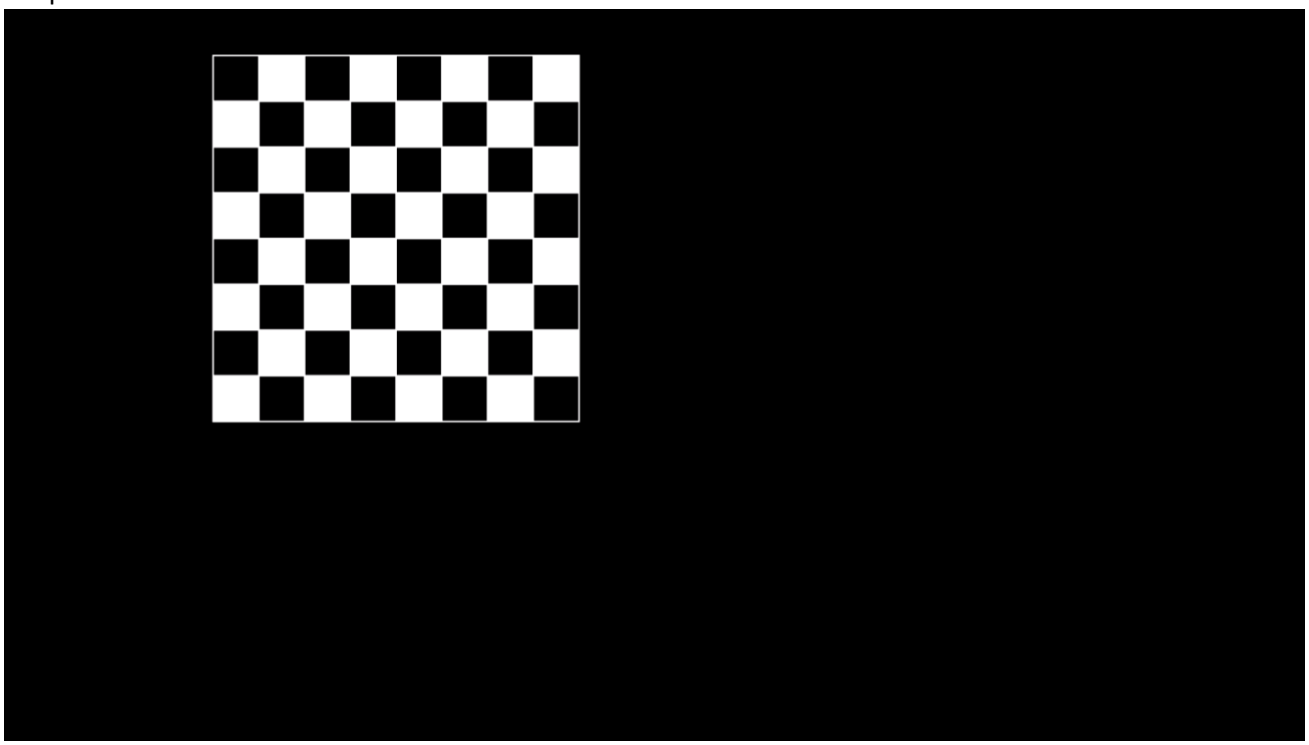
Assignment 7

DEV SHAH(202000362)

Code:

```
// C program to create a chess board
#include <conio.h>
#include <dos.h>
#include <graphics.h>
#include <stdio.h>

void main()
{

	int gr = DETECT, gm;
	int r, c, x = 30, y = 30, black = 0;
	for (r = 0; r < 8; r++) {
		for (c = 1; c <= 8; c++)
{
			if (black == 0) {

				setcolor(WHITE);
				setfillstyle(SOLID_FILL, BLACK);
				rectangle(x, y, x + 30, y + 30);
				floodfill(x + 1, y + 1, WHITE);
				black = 1;
			}

		else {
				setcolor(WHITE);
				setfillstyle(SOLID_FILL, WHITE);
				rectangle(x, y, x + 30, y + 30);
				floodfill(x + 1, y + 1, WHITE);
				black = 0;
			}
			x = x + 30;

			delay(30);
		}
		if (black == 0)
			black = 1;
		else
			black = 0;

		delay(30);
		x = 30;
		y = 30 + y;
	}
	getch();

	closegraph();
}
```

Output:



Name: Dev DInesh Shah

SRN: 202000362

Subject : Social Media Analytics Lab

LAB 9

Q)write a program in C to apply 3D transformation on a given object.

SOURCE CODE:

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
#include<math.h>
int maxx,maxy,midx,midy;
void axis()
{
getch();
cleardevice();
line(midx,0,midx,maxy);
line(0,midy,maxx,midy);
}
void main()
{
int gd,gm,x,y,z,ang,x1,x2,y1,y2;
detectgraph(&gd,&gm);
initgraph(&gd,&gm,"C://TURBOC3//BGI");
setfillstyle(3,25);
maxx=getmaxx();
```

```
maxy=getmaxy();
midx=maxx/2;
midy=maxy/2;
outtextxy(100,100,"ORIGINAL OBJECT");
line(midx,0,midx,maxy);
line(0,midy,maxx,midy);
bar3d(midx+100,midy-20,midx+60,midy-90,20,5);
axis();
outtextxy(100,20,"TRANSLATION");
printf("\n\n Enter the Translation vector: ");
scanf("%d%d",&x,&y);
bar3d(midx+100,midy-20,midx+60,midy-90,20,5);
bar3d(midx+(x+100),midy-(y+20),midx+(x+60),midy-(y+90),20
,5);
axis();
outtextxy(100,20,"SCALING");
printf("\n Enter the Scaling Factor: ");
scanf("%d%d%d",&x,&y,&z);
bar3d(midx+100,midy-20,midx+60,midy-90,20,5);
bar3d(midx+(x*100),midy-(y*20),midx+(x*60),midy-(y*90),20*
z,5);
axis();
outtextxy(100,20,"ROTATION");
```

```
printf("\n Enter the Rotation angle: ");
scanf("%d",&ang);
x1=100*cos(ang*3.14/180)-20*sin(ang*3.14/180);
y1=100*sin(ang*3.14/180)+20*sin(ang*3.14/180);
x2=60*cos(ang*3.14/180)-90*sin(ang*3.14/180);
y2=60*sin(ang*3.14/180)+90*sin(ang*3.14/180);
axis();
printf("\n After rotating about z-axis\n");
bar3d(midx+100,midy-20,midx+60,midy-90,20,5);
bar3d(midx+x1,midy-y1,midx+x2,midy-y2,20,5);
axis();
printf("\n After rotating about x-axis\n");
bar3d(midx+100,midy-20,midx+60,midy-90,20,5);
bar3d(midx+100,midy-x1,midx+60,midy-x2,20,5);
axis();
printf("\n After rotating about y-axis\n");
bar3d(midx+100,midy-20,midx+60,midy-90,20,5);
bar3d(midx+x1,midy-20,midx+x2,midy-90,20,5);
axis();
closegraph();
}
```

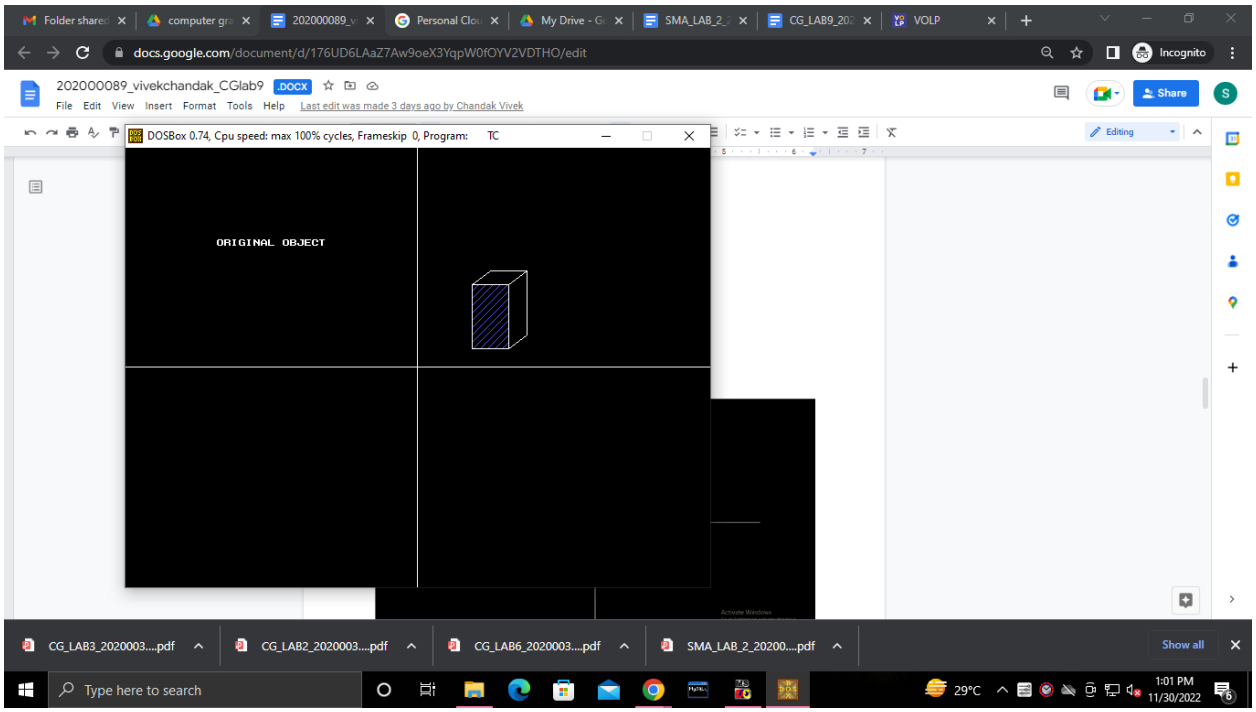
Output:

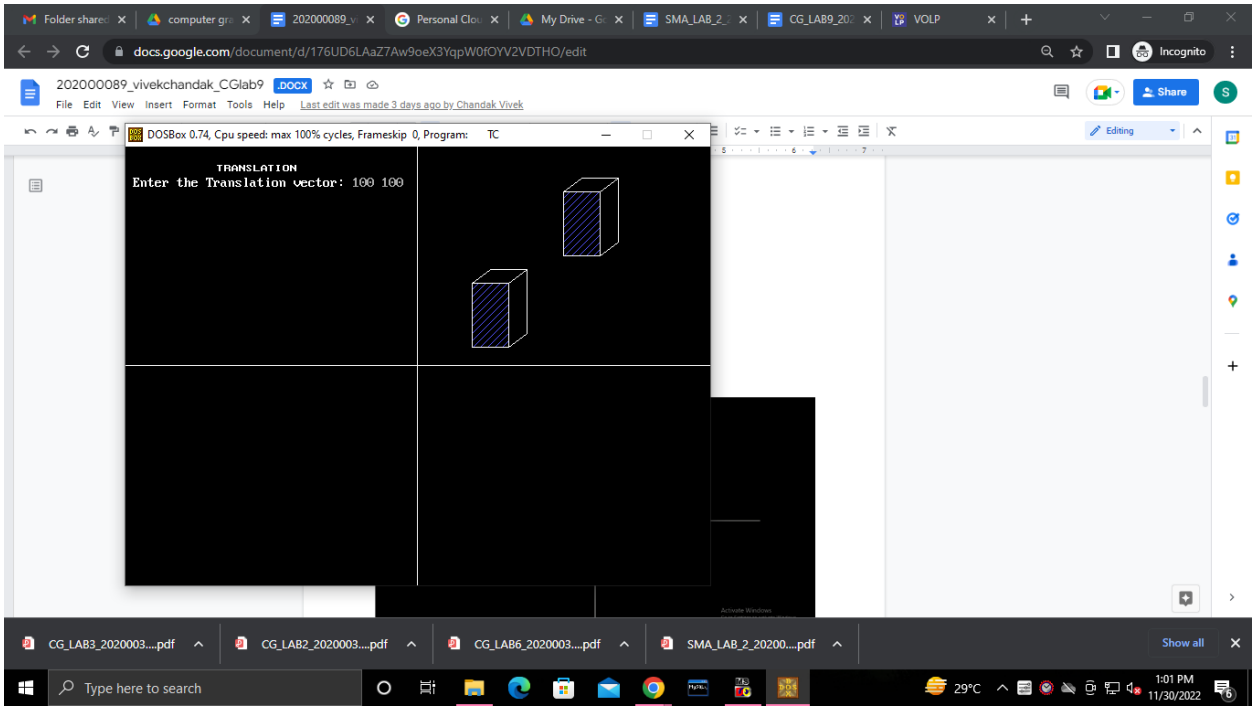
Input:

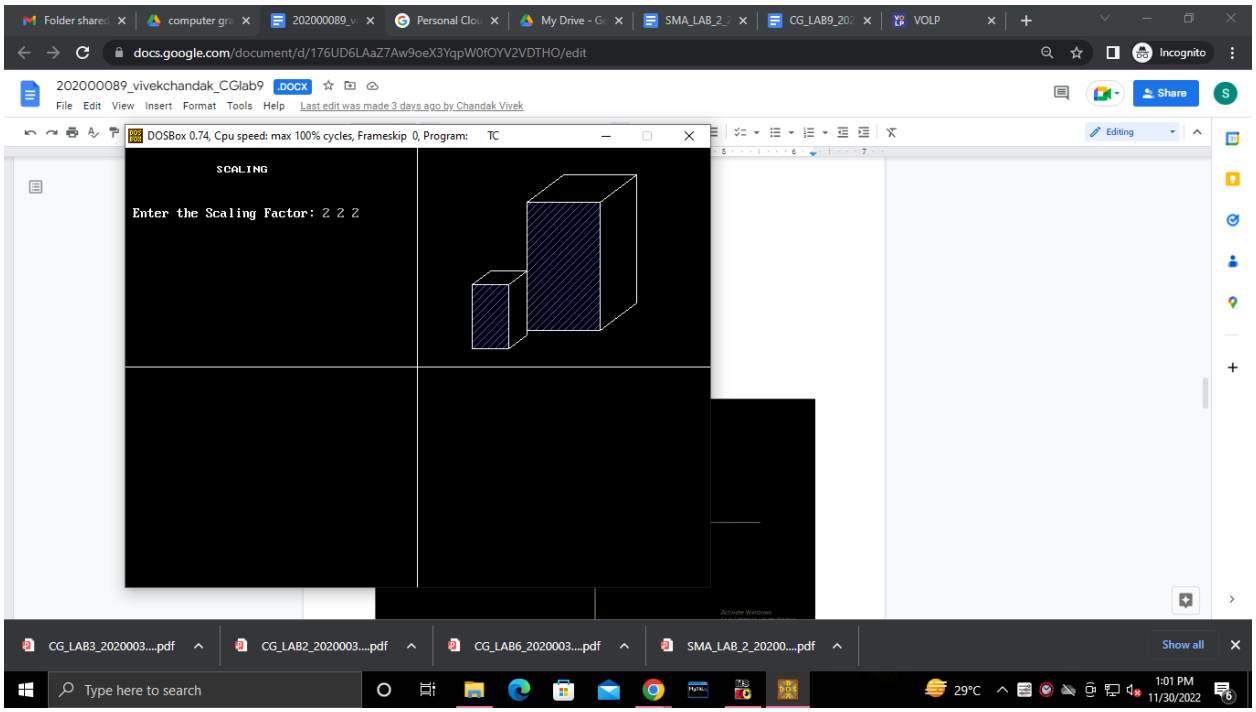
100 100 100

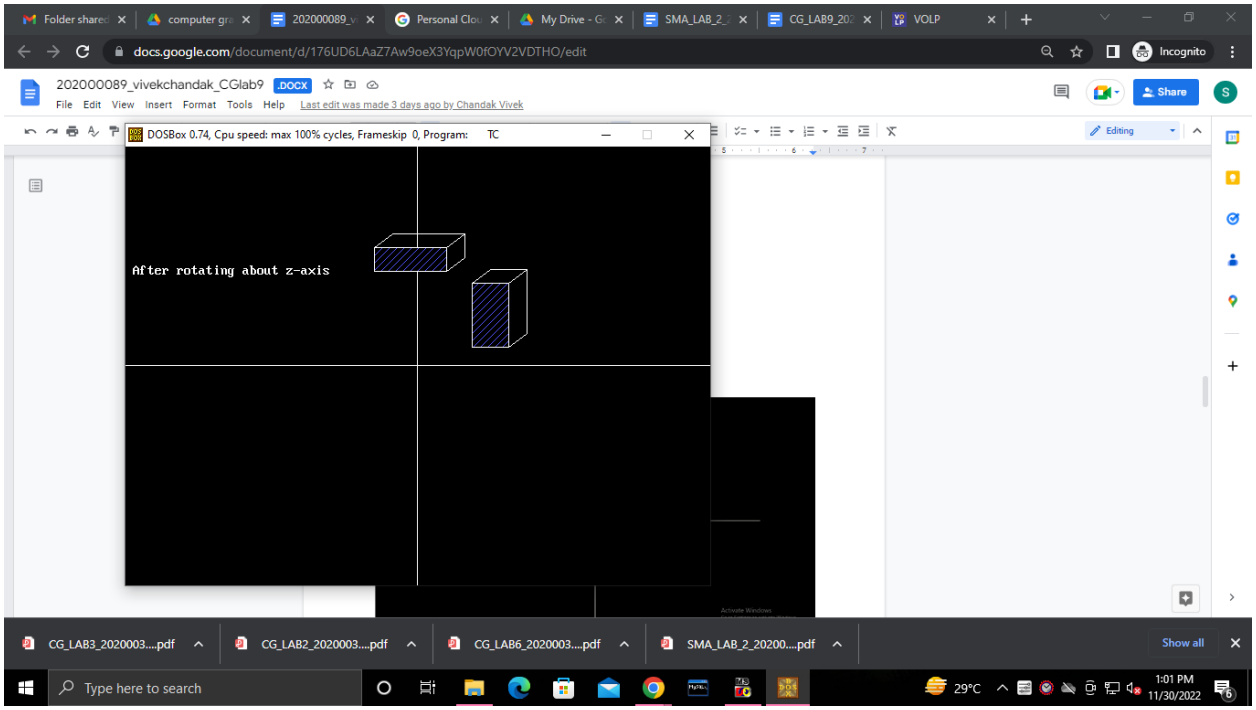
2 2 2

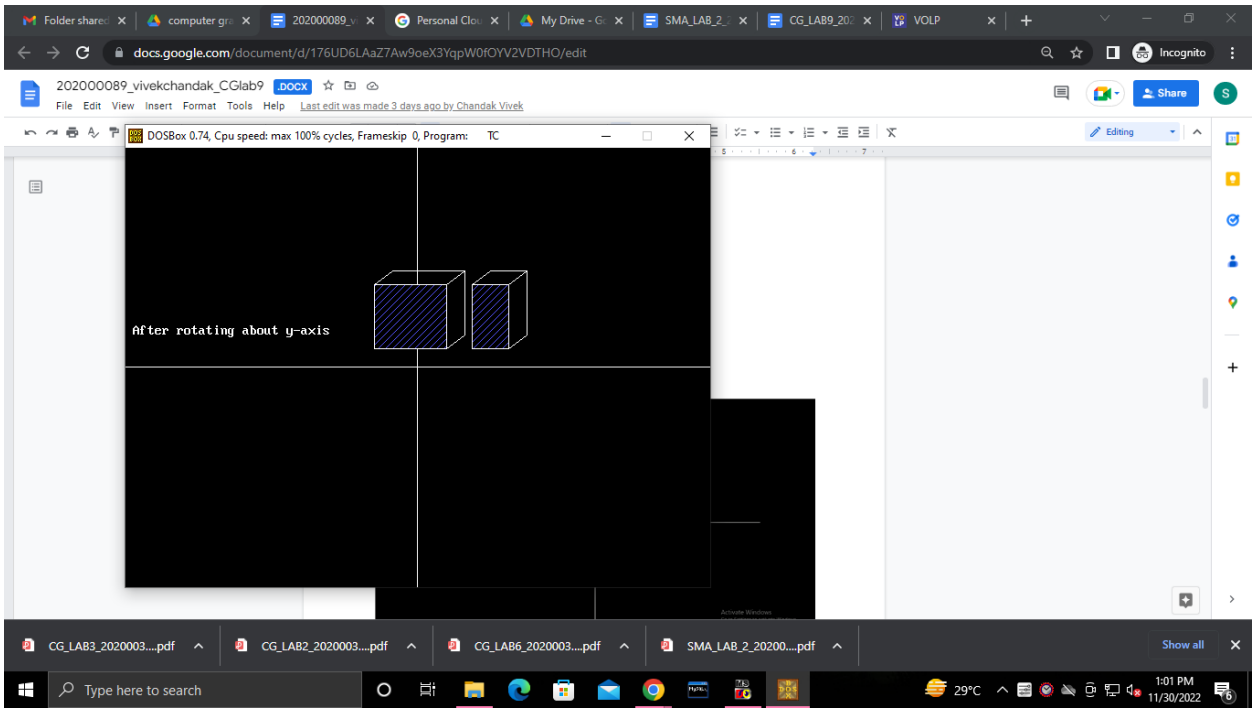
60

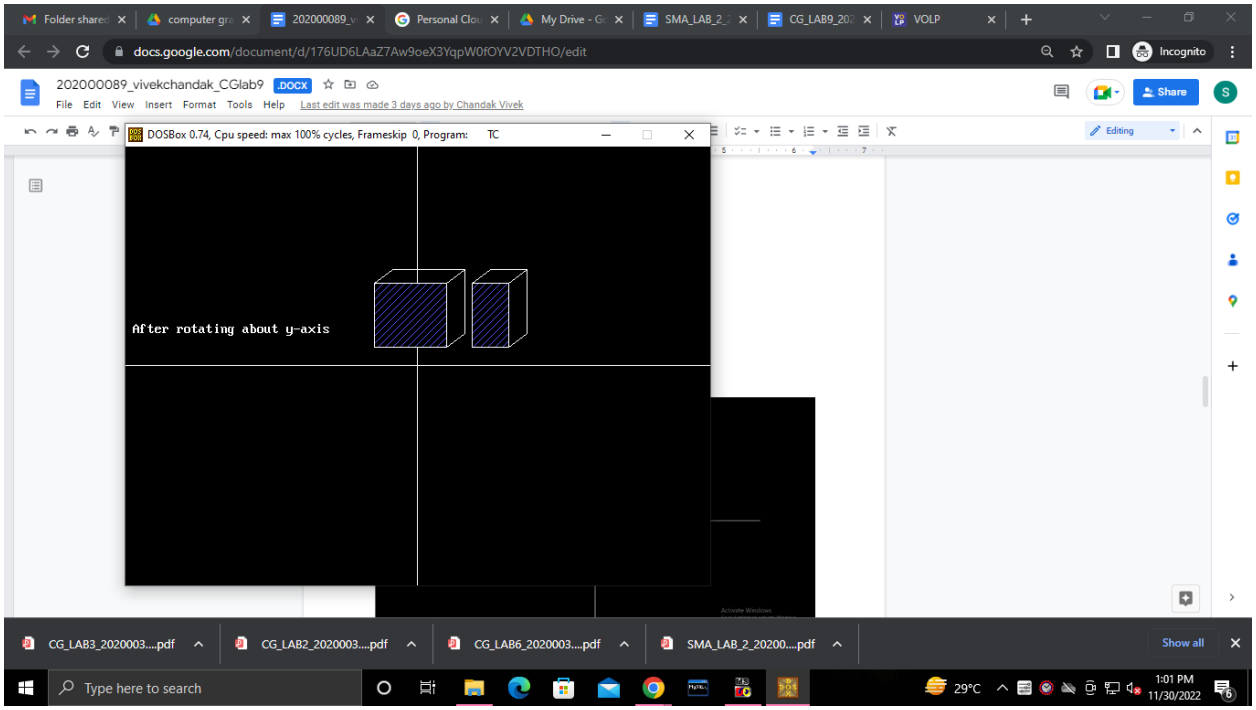



docs.google.com/document/d/176UD6LAaZ7Aw9oeX3YqpW0fOYV2VDTHO/edit
202000089_vivekchandak_CGlab9 .DOCX
File Edit View Insert Format Tools Help Last edit was made 3 days ago by Chandak Vivek
DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC
TRANSLATION
Enter the Translation vector: 100 100
CG_LAB3_2020003....pdf
CG_LAB2_2020003....pdf
CG_LAB6_2020003....pdf
SMA_LAB_2_20200....pdf
Show all
Type here to search
29°C
1:01 PM
11/30/2022


docs.google.com/document/d/176UD6LAaZ7Aw9oeX3YqpW0fOYV2VDTHO/edit
202000089_vivekchandak_CGlab9 .DOCX
File Edit View Insert Format Tools Help Last edit was made 3 days ago by Chandak Vivek
DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC
SCALING
Enter the Scaling Factor: 2 2 2
CG_LAB3_2020003....pdf
CG_LAB2_2020003....pdf
CG_LAB6_2020003....pdf
SMA_LAB_2_20200....pdf
Show all
Type here to search
29°C
1:01 PM
11/30/2022


DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC
After rotating about z-axis


DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC
After rotating about y-axis


DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC
After rotating about y-axis