



DEV SANSKRITI
VISHWAVIDYALAYA



DEV SANSKRITI VISHWAVIDYALAYA

SESSION 2018-21

**Practical File
Of**

Computer Graphics using C

SUBMITTED TO:

Mr. Swapnil Gaidhani
Lecturer

SUBMITTED BY:

Aniket Kumar
BCA (4th Sem)

Department of Computer Science,
DSVV, Haridwar

INDEX

Task No.	Description	Page	Remark
1	Write a program in C to display the screen resolution	1	
2	Draw primitive shapes viz., circle, rectangle, line, ellipse, sector, and polygon	2	
3	Draw a grid.	4	
4	Draw a bar graph.	6	
5	Draw a pie chart.	8	
6	Draw a face using primitive shapes.	10	
7	Draw shapes within shapes viz., concentric circles, box within, and triangles within triangles.	12	
8	Draw a line using the DDA (Digital Differential Analyzer) algorithm. Draw a line using the Bresenham's Line algorithm.	15 18	
9	Draw a circle using Mid point circle generation algorithm.	21	
10	Draw any character using the Stroke method.	24	
11	Draw a character using the Bitmap method.	25	
12	Write a program using C to demonstrate Flood fill and boundary fill algorithms	27	
13	Demonstrate 2D clipping :- Cohen-Sutherland line clipping.	31	
14	Demonstrate rotation of any shape.	37	
15	Demonstrate translation of any shape.	40	
16	Develop any 2D Game of your choice	45	
17	Draw a line using basic mathematical calculations ($y=mx+c$)	85	
18	Draw a circle using trigonometric functions	88	
19	Draw a circle and move it according to the press of arrow keys.	90	
20	Write a program in C to demonstrate 2D Transformation : Scaling	93	

Task 1

Write a program in C to display the screen resolution

```
// A program in C to display the resolution of the Screen.
#include<graphics.h>
#include<stdio.h>
#include<conio.h>

void main()
{
    int gd=DETECT,gm,x,y;
    initgraph(&gd,&gm,"c:\\turbo3\\bgi");

    x=getmaxx();    //maximum coordinate in the x direction
    y=getmaxy();    //maximum coordinate in the y direction
    printf("Resolution of the screen is :  %d * %d Pixels",x,y);
    getch();
    closegraph();
}
```

Screenshot :

A screenshot of a black terminal window with white text. The text reads: "Resolution of the screen is : 639 * 479 Pixels".

Resolution of the screen is : 639 * 479 Pixels

Task 2

Draw primitive shapes viz., circle, rectangle, line, ellipse, sector, and polygon.

```
/* A program in C to draw some primitive shapes viz. circle,
rectangle, line, ellipse, sector and polygon */
```

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

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

```
#include<conio.h>
```

```
void main()
```

```
{
```

```
    int gd=DETECT,gm;
```

```
    int arr[]={340,150,470,180,400,240,340,150};
```

```
//coordinates for the polygon
```

```
    initgraph(&gd,&gm,"c:\\turboc3\\bgi");
```

```
// iniatilizes the graphics driver
```

```
    circle(50,60,50);    //(x,y,radius)
```

```
    outtextxy(25,120,"Circle");
```

```
    rectangle(150,20,300,100);    //(left top, right bottom)
```

```
    outtextxy(180,120,"Rectangle");
```

```
    line(350,30,480,80);    //(x1,y1,x2,y2)
```

```
    outtextxy(400,120,"Line");
```

```
    ellipse(100,200,0,360,80,40);
```

```
x-mid,y-mid,st_angle,end angle,x-radius,y-radius)
```

```
    outtextxy(70,260,"Ellipse");
```

```
    sector(250,200,0,240,50,50);
```

```
    //(x-mid,y-mid,st_angle,end angle,x-radius,y-radius)
```

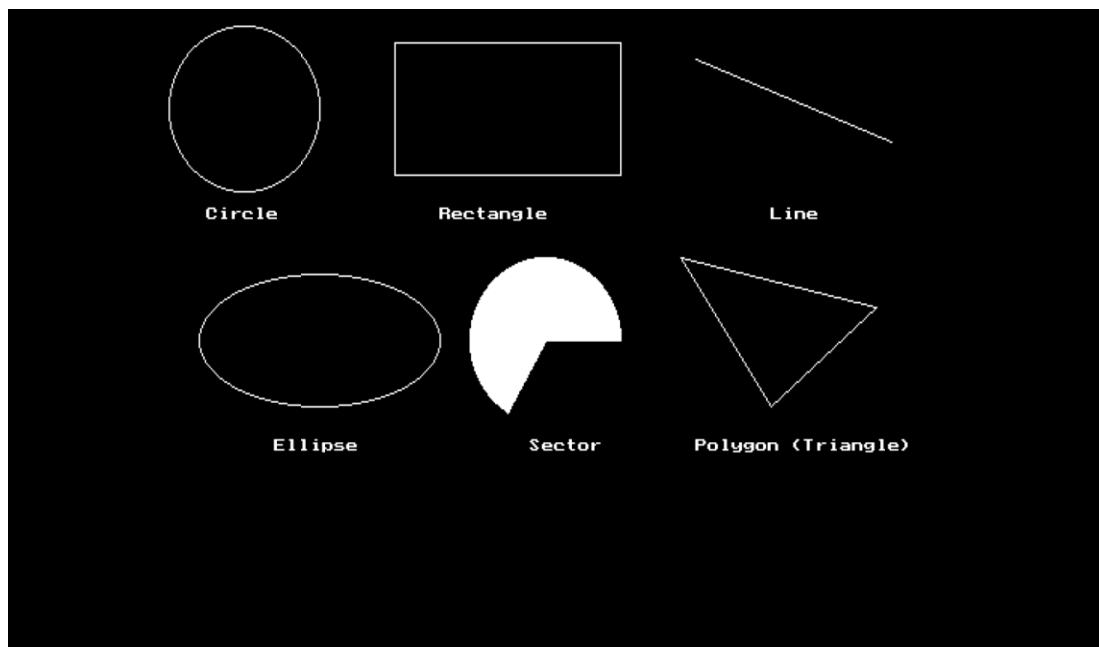
```

    outtextxy(240,260,"Sector");
    drawpoly(4,arr);          //(no of points, name of array)
    outtextxy(350,260,"Polygon (Triangle)");

    getch();
    closegraph();
}

```

Screenshot :



Task 3

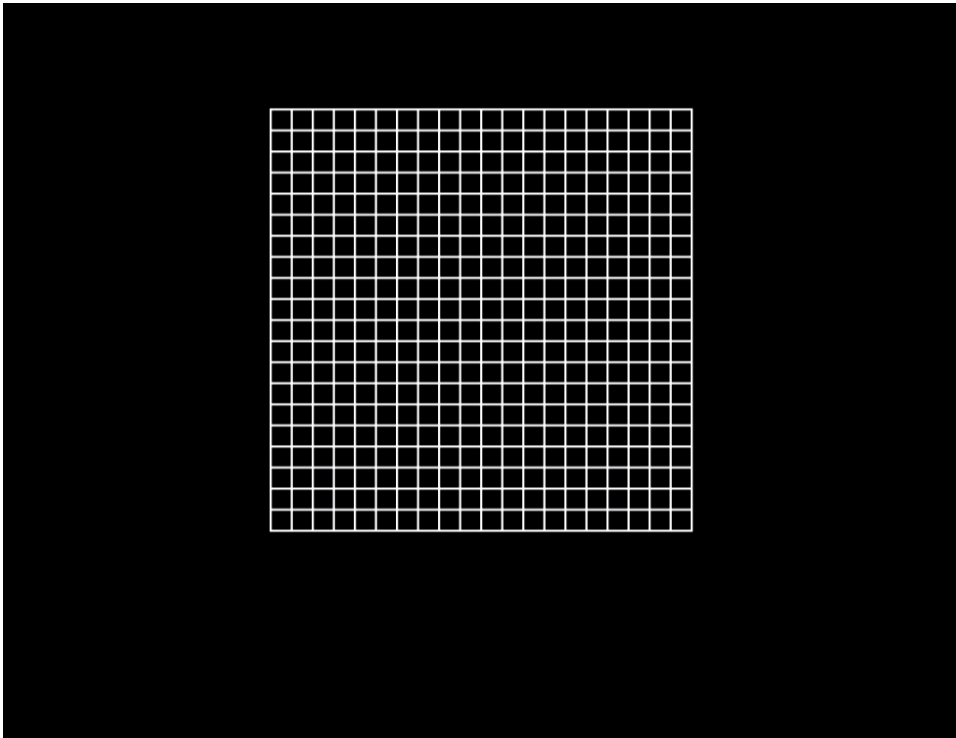
Draw a grid.

```
// A program in C to draw a grid.
#include<graphics.h>
#include<stdio.h>
#include<conio.h>

void main()
{
    int gd=DETECT,gm,i,j;
    initgraph(&gd,&gm,"c:\\turbo3\\bgi");

    for(i=0;i<=200;i+=10)
    {
        line(20+i,50,20+i,250); //vertical lines
    }
    for(j=0;j<=200;j+=10)
    {
        line(20,50+j,220,50+j); //horizontal lines
    }
    getch();
    closegraph();
}
```

Screenshot :



Task 4

Draw a bar graph.

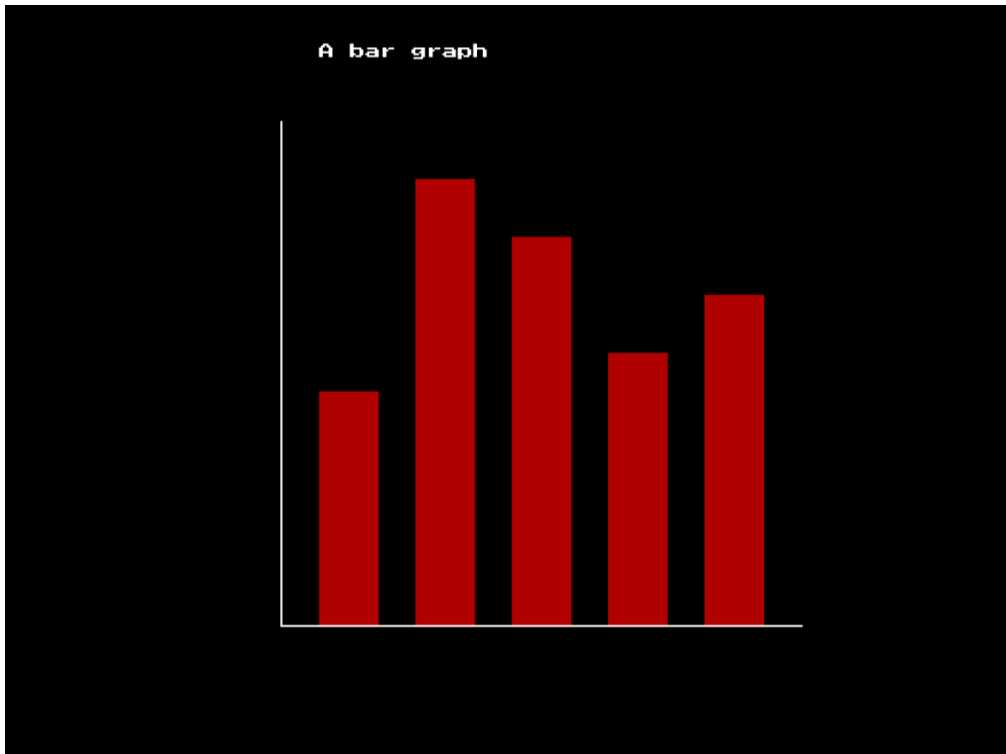
```
// A program in C to draw a bar graph.
#include<graphics.h>
#include<stdio.h>
#include<conio.h>

void main()
{
    int gd=DETECT,gm,x,y;
    initgraph(&gd,&gm,"c:\\turbo3\\bgi");

    outtextxy(150,20,"A bar graph");
    line(130,321,400,321); //x-axis
    line(130,321,130,60); //y-axis

    setfillstyle(SOLID_FILL,RED); //filling style and colour
    bar(150,200,180,320);
    bar(200,90,230,320);
    bar(250,120,280,320);
    bar(300,180,330,320);
    bar(350,150,380,320);
    getch();
    closegraph();
}
```


Screenshot :



Task 5

Draw a pie chart.

```
// A program in C to draw a pie chart.
#include<graphics.h>
#include<stdio.h>
#include<conio.h>

void main()
{
    int gd=DETECT,gm;
    initgraph(&gd,&gm,"c:\\turbo3\\bgi");

    outtextxy(150,100,"A pie chart");

    setfillstyle(SOLID_FILL,RED); //filling style and colour
    pieslice(200,200,0,150,80);    //150 degree
    outtextxy(200,200-40,"41.7%");

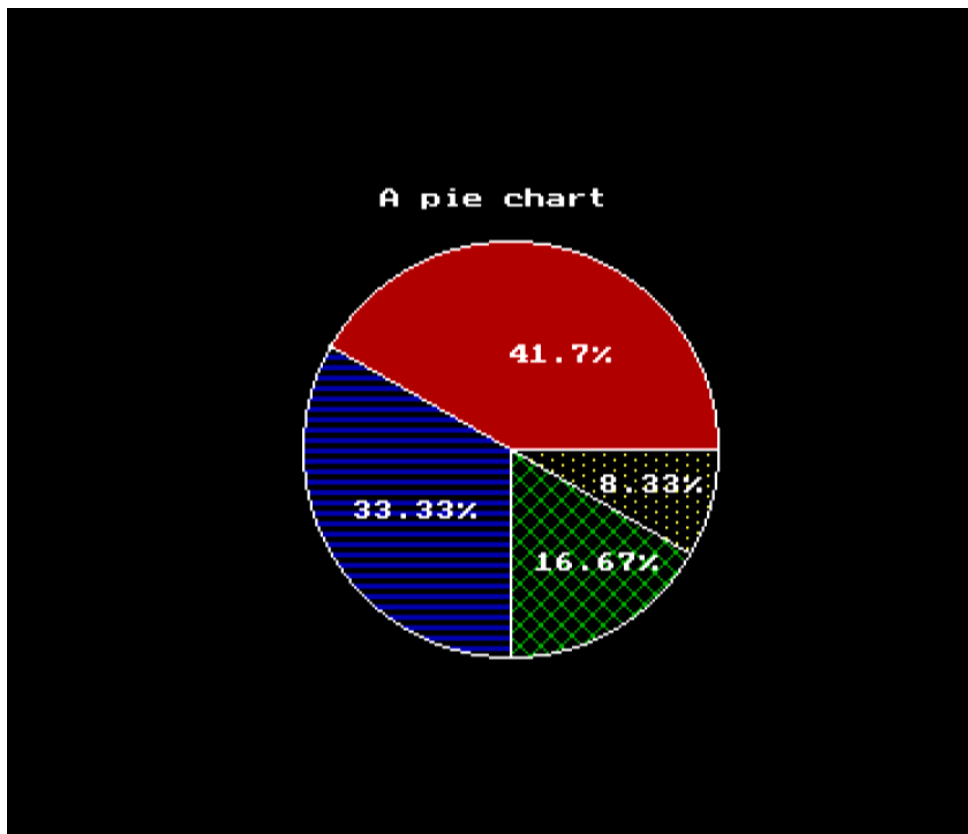
    setfillstyle(LINE_FILL,BLUE);
    pieslice(200,200,150,270,80); //120 degree
    outtextxy(200-60,200+20,"33.33%");

    setfillstyle(XHATCH_FILL,GREEN);
    pieslice(200,200,270,330,80); //60 degree
    outtextxy(200+10,200+40,"16.67%");

    setfillstyle(WIDE_DOT_FILL,YELLOW);
    pieslice(200,200,330,360,80); //30 degree
    outtextxy(200+35,200+10,"8.33%");
```

```
    getch() ;  
    closegraph() ;  
  
}
```

Screenshot :



Task 6

Draw a face using primitive shapes.

```
// A program in C to draw a face using primitive shapes.
#include<graphics.h>
#include<stdio.h>
#include<conio.h>

void main()
{
    int gd=DETECT,gm,x,y;
    initgraph(&gd,&gm,"c:\\turbo3\\bgi");

    outtextxy(90,100,"A face using primitive shapes");
    setcolor(YELLOW);
    circle(200,200,80);    //face
    ellipse(170,180,0,360,12,8);    //left eye
    ellipse(230,180,0,360,12,8);    //right eye
    line(200-5,195,200+5,230);    //nose
    ellipse(198,230,0,360,8,3);
    arc(200,215,220,320,40);    //mouth
    arc(200,265,40,135,20);    //tongue
    getch();
    closegraph();
}
```

Screenshot :



Task 7

Draw shapes within shapes viz., concentric circles, box within, and triangles within triangles.

```
// A program to draw concentric shapes.
#include<stdio.h>
#include<conio.h>
#include<graphics.h>

void mycircle();
void myrectangle();
void mytriangle();
void main()
{
    int gd=DETECT,gm;
    initgraph(&gd,&gm,"c:\\turbo3\\bgi");

    outtext("This is a program to draw shapes within shapes");
    //calling the 3 shapes
    mycircle();
    myrectangle();
    mytriangle();

    getch();
    closegraph();
}

void mycircle()
{
    int i,x=80,y=100,r=10;
    setcolor(RED); //sets the outline colour RED
```

```

    for(i=0;i<=60;i+=10)
    //each circle having a difference of radius 10p
    {
        circle(x,y,r+i);
    }
}

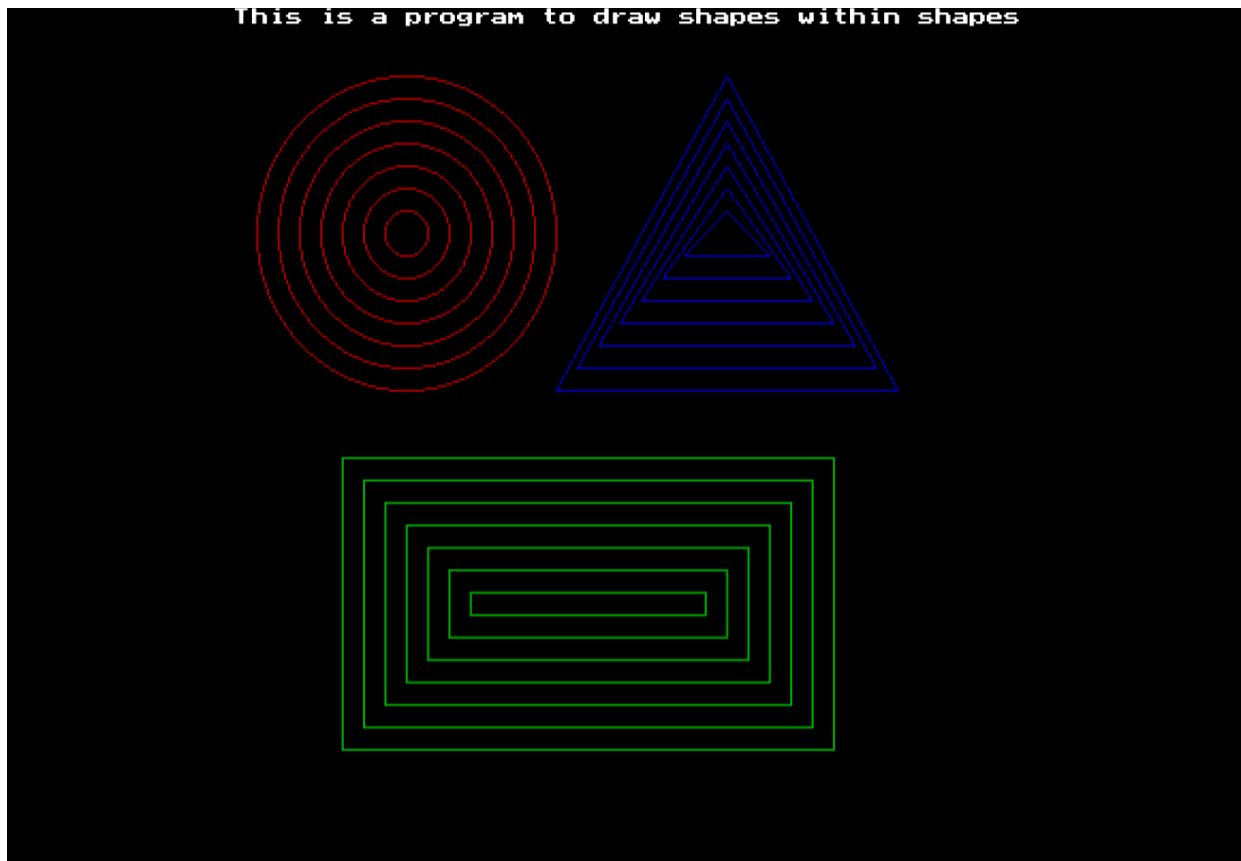
void myrectangle()
{
    int i,x1=50,y1=200,x2=280,y2=330;
    setcolor(GREEN); //sets the outline colour GREEN

    for(i=0;i<=60;i+=10)
    //each rectangle having a difference of length 10p
    {
        rectangle(x1+i,y1+i,x2-i,y2-i);
    }
}

void mytriangle()
{
    int i,x1=230,y1=30,x2=310,y2=170,x3=150,y3=170;
    setcolor(BLUE); //sets the outline colour GREEN
    for(i=0;i<=60;i+=10)
    //each rectangle having a difference of length 10p
    {
        line(x1,y1+i,x2-i,y2-i); //right arm of the triangle
        line(x2-i,y2-i,x3+i,y3-i); //base of the triangle
        line(x3+i,y3-i,x1,y1+i); //left arm of the triangle
    }
}

```

Screenshot :



Task 8(i)

Draw a line using the DDA (Digital Differential Analyzer) algorithm.

```
// A program to print a line using DDA (Digital Differential
Algorithm) Algorithm.

#include<stdio.h>
#include<conio.h>
#include<graphics.h>
#include<stdlib.h>

void dda(float, float, float, float);
main()
{
    float x0,x1,y0,y1;
    clrscr();
    {
        printf("Enter the value of x1 and y1 :\n");
        scanf("%f%f",&x0,&y0);
        printf("\n\nEnter the value of x2 and y2 : \n");
        scanf("%f%f",&x1,&y1);
        dda(x0,x1,y0,y1);
    }
}

void dda(float x0,float x1,float y0,float y1)
{
    int gd=DETECT,gm;
    float dx,dy,i,xinc,yinc,step;

    dx=(x1-x0); //difference between x-coordinates
    dy=(y1-y0); //difference between y-coordinates
```

```

//assigning the value of step
if (abs(dx)>abs(dy))
{
    step=abs(dx);
}
else
{
    step=abs(dy);
}
xinc=dx/step; //increment in x-coordinate
yinc=dy/step; //increment in y-coordinate

initgraph(&gd,&gm,"c:\\turbo3\\bgi");
outtext("Line drawn using DDA Algorithm.");

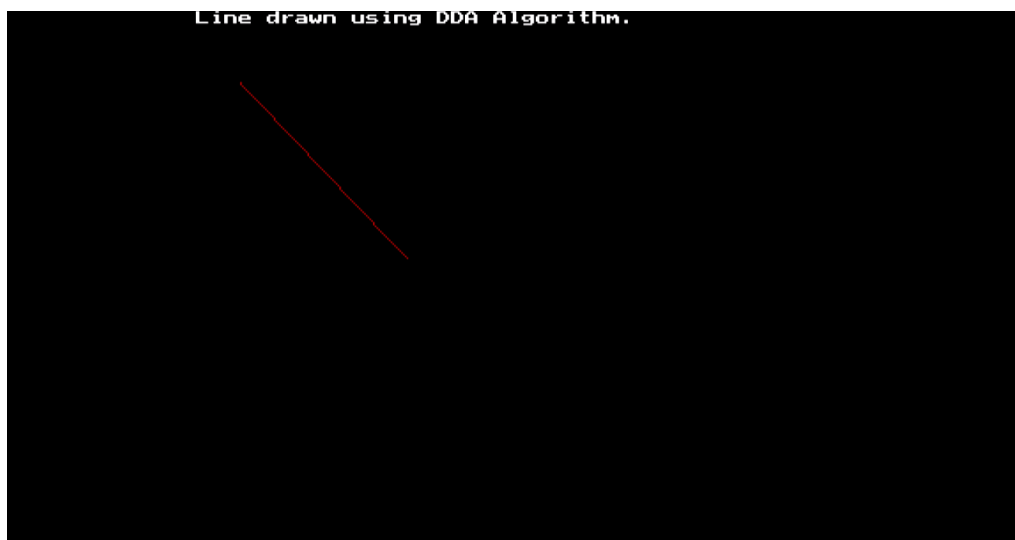
for(i=0;i<step;i++)
{
    putpixel(x0,y0,RED);
    x0=x0+xinc;
    y0=y0+yinc;
}

//line(x0,y0,x1,y1);
getch();
closegraph();
}

```

Screenshots :

```
Enter the value of x1 and y1 :  
25  
40  
  
Enter the value of x2 and y2 :  
120  
140
```



Task 8(ii)

Draw a line using the Bresenham's Line algorithm.

```
// A program to print a line using Bressenhem's Line Algorithm
#include<stdio.h>
#include<conio.h>
#include<graphics.h>

void bla(int, int, int, int);
main()
{
    int x1,x2,y1,y2;
    clrscr();
    {
        printf("Enter the value of x1 and y1 :\n");
        scanf("%d%d",&x1,&y1);
        printf("\n\nEnter the value of x2 and y2 : \n");
        scanf("%d%d",&x2,&y2);
        bla(x1,y1,x2,y2);
    }
}

void bla(int x1, int y1, int x2, int y2)
{
    int gd=DETECT,gm;
    int dx,dy,dp;

    dy=y2-y1;    //difference in y-coordinates
    dx=x2-x1;    //difference in x-coordinates

    initgraph(&gd,&gm,"c:\\\\turboc3\\\\bgi");
```

```

//line(x1,y1,x2,y2);
outtext("Line drawn using Bressenhem's Line Algorithm.");

dp=2*dy-dx;    //decision parameter

while(x1<x2)
{
    if(dp>=0)
    {
        putpixel(x1,y1,RED);
        y1=y1+1;
        dp=dp+2*dy-2*dx;
    }
    else
    {
        putpixel(x1,y1,RED);
        dp=dp+2*dy;
    }
    x1=x1+1;
}
getch();
closegraph();
}

```

Screenshots :

```
Enter the value of x1 and y1 :
```

```
20
```

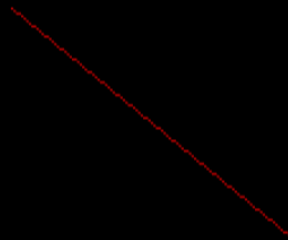
```
30
```

```
Enter the value of x2 and y2 :
```

```
130
```

```
120_
```

```
Line drawn using Bresenham's Line Algorithm.
```



Task 9

Draw a circle using Mid point circle generation algorithm.

```
// A program to demonstrate midpoint circle algorithm
#include<stdio.h>
#include<conio.h>
#include<graphics.h>

void midpoint_circle(int, int, int);
void main()
{
    int gd=DETECT, gm;
    int x,y,r;

    //getting the coordinates and radius as input
    printf("This a program to demonstrate Midpoint Circle
Algorithm");
    printf("\nEnter the coordinates of the center (X and Y):\n");
    scanf("%d%d",&x,&y);
    printf("\nEnter the radius of circle: ");
    scanf("%d",&r);

    initgraph(&gd,&gm,"c:\\turbo3\\bgi");
    midpoint_circle(x,y,r); //calling the required function

    getch();
    closegraph();
}

void midpoint_circle(int x, int y, int r)
{

```

```

int s=0,dp=0; //decision parameters

while(r>=s)
{
    // 8 midpoints of the circle, calling recursively
    putpixel(x+r,y+s,RED);
    putpixel(x+s,y+r,RED);
    putpixel(x-s,y+r,RED);
    putpixel(x-r,y+s,RED);
    putpixel(x-r,y-s,RED);
    putpixel(x-s,y-r,RED);
    putpixel(x+s,y-r,RED);
    putpixel(x+r,y-s,RED);

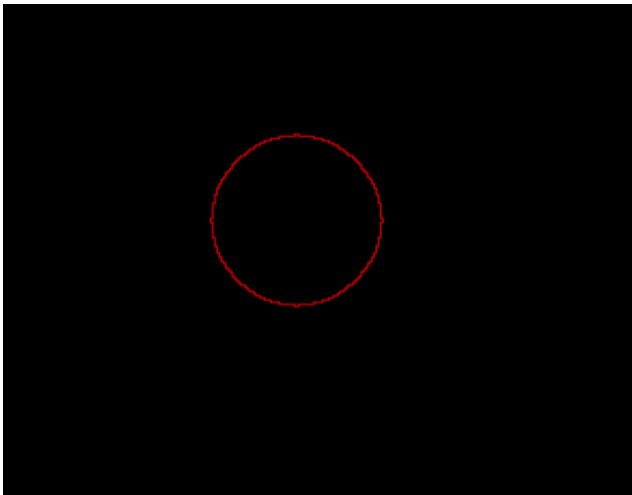
    //conditions for decision parameters
    if(dp<=0)
    {
        s=s+1; //increasing the value after 1 iteration
        dp=dp+2*s+1;
    }
    else
    {
        r=r-1; //increasing the value after 1 iteration
        dp=dp-2*r+1;
    }
}
}

```


Screenshots :

```
C:\TURBOC3\BIN>TC
This a program to demonstrate Midpoint Circle Algorithm
Enter the coordinates of the center (X and Y):
100 100

Enter the radius of circle: 40
```



Task 10

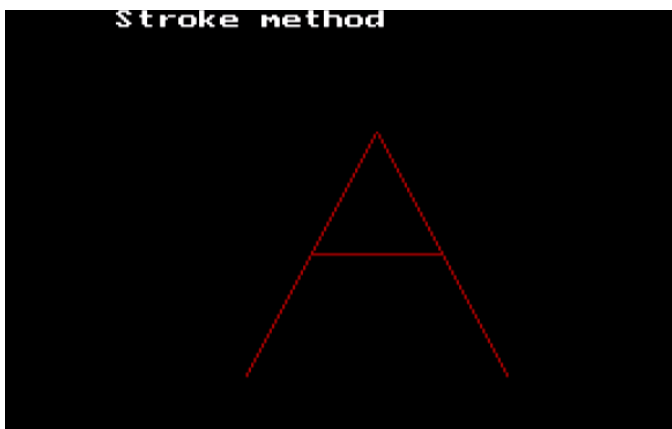
Draw any character using the Stroke method.

```
// A program to draw a character (A), using Stroke method.
#include<stdio.h>
#include<conio.h>
#include<graphics.h>

void main()
{
    int gd=DETECT,gm;
    initgraph(&gd,&gm,"c:\\turbo3\\bgi");

    outtext("Stroke method");
    setcolor(RED);
    line(100,50,50,150);
    line(100,50,150,150);
    line(75,100,125,100);
    getch();
    closegraph();
}
```

Screenshot :



Task 11

Draw a character using the Bitmap method.

```
// A program to draw a character (A), using Bitmap method.
#include<stdio.h>
#include<conio.h>
#include<graphics.h>

void main()
{
    int gd=DETECT,gm,i,j,c=2;  //c is for colour
    int a[14][9]={0,0,0,0,1,0,0,0,0},
                {0,0,0,1,1,1,0,0,0},
                {0,0,1,1,0,1,1,0,0},
                {0,1,1,0,0,0,1,1,0},
                {1,1,0,0,0,0,0,1,1},
                {1,1,0,0,0,0,0,1,1},
                {1,1,0,0,0,0,0,1,1},
                {1,1,1,1,1,1,1,1,1},
                {1,1,0,0,0,0,0,1,1},
                {1,1,0,0,0,0,0,1,1},
                {1,1,0,0,0,0,0,1,1},
                {1,1,0,0,0,0,0,1,1},
                {1,1,0,0,0,0,0,1,1},
                {1,1,0,0,0,0,0,1,1},
                {1,1,0,0,0,0,0,1,1},
    };

    initgraph(&gd,&gm,"c:\\turbo3\\bgi");

    outtext("Bitmap method");
    for(i=0;i<14;i++)
```

```
{  
    for(j=0;j<9;j++)  
    {  
        putpixel(50+j,20+i,c*a[i][j]);  
    }  
}  
getch();  
closegraph();  
}
```

Screenshot :



Task 12

Write a program using C to demonstrate Flood fill and boundary fill algorithms

```
// A program to demonstrate flood fill.
#include<stdio.h>
#include<conio.h>
#include<graphics.h>

void flood_fill(int, int, int, int);

void main()
{
    int gd=DETECT,gm;
    int x1=50,y1=50,x2=120,y2=100;
    //coordinates of the rectangle
    int x0=x1+20,y0=y1+20;    //coordinate, from where to start
    int bk_col=0, fill_col=1;
    //old colour and colour to be filled

    initgraph(&gd,&gm,"c:\\turboc3\\bgi");

    outtext("This is Flood Fill.");
    rectangle(x1,y1,x2,y2);
    flood_fill(x0,y0,bk_col,fill_col);
    //calling the recursive function

    getch();
    closegraph();
}
```

```

void flood_fill(int x, int y, int bk_col, int fill_col)
{
    if(getpixel(x,y)==bk_col)
        //checks for the old/background colour
    {
        delay(2);
        putpixel(x,y,fill_col); //the starting pixel
        flood_fill(x+1,y,bk_col,fill_col); //right pixel
        flood_fill(x-1,y,bk_col,fill_col); //left pixel
        flood_fill(x,y+1,bk_col,fill_col); //bottom pixel
        flood_fill(x,y-1,bk_col,fill_col); //upward pixel
        flood_fill(x+1,y+1,bk_col,fill_col); //bottom right px
        flood_fill(x-1,y+1,bk_col,fill_col); //bottom left px
        flood_fill(x+1,y-1,bk_col,fill_col); //top right pixel
        flood_fill(x-1,y-1,bk_col,fill_col); //top left pixel
    }
}

```

Screenshot :



```

// A program to demonstrate boundary fill.

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

void boundary_fill(int, int, int, int);

void main()
{
    int gd=DETECT,gm;
    int x1=50,y1=50,x2=120,y2=120;
    //coordinates of the rectangle
    int x0=x1+20,y0=y1+20;    //coordinate, from where to start
    int bound_col=15, fill_col=5;
    //old colour and colour to be filled

    initgraph(&gd,&gm,"c:\\turboc3\\bgi");

    outtext("This is Boundary fill.");
    rectangle(x1,y1,x2,y2);
    boundary_fill(x0,y0,bound_col,fill_col);
    //calling the recursive function

    getch();
    closegraph();
}

void boundary_fill(int x, int y, int bound_col, int fill_col)
{
    if(getpixel(x,y)!=bound_col && getpixel(x,y)!=fill_col)
    {

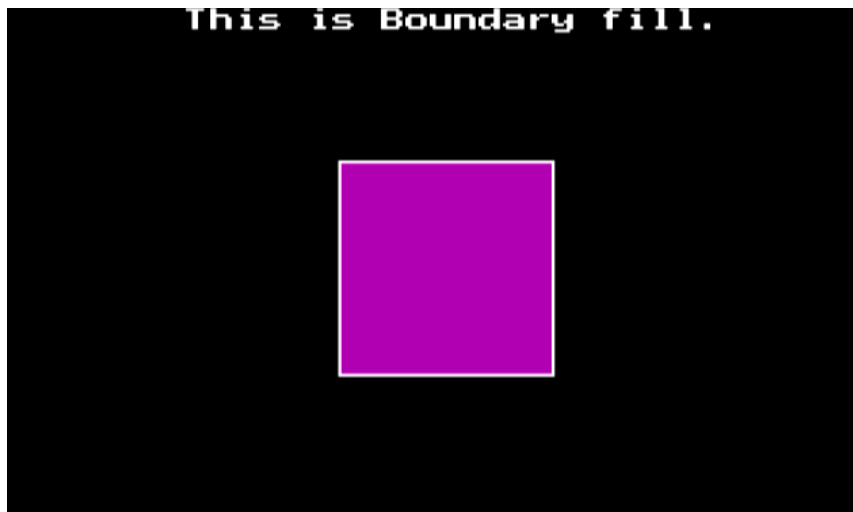
```

```

delay(2);
putpixel(x,y,fill_col);//starting pixel
boundary_fill(x+1,y,bound_col,fill_col);//right pixel
boundary_fill(x-1,y,bound_col,fill_col);//left pixel
boundary_fill(x,y+1,bound_col,fill_col);//bottom pixel
boundary_fill(x,y-1,bound_col,fill_col);//upward pixel
boundary_fill(x+1,y+1,bound_col,fill_col);//btm right p
boundary_fill(x-1,y+1,bound_col,fill_col);//btm lft p
boundary_fill(x+1,y-1,bound_col,fill_col);//top right p
boundary_fill(x-1,y-1,bound_col,fill_col);//top left p
}
}

```

Screenshot :



Task 13

Demonstrate 2D clipping :- Cohen-Sutherland line clipping.

```
/* This is a program in C to demonstrate 2D clipping using Cohen-
Sutherland line clipping */

#include<conio.h>

#include<stdio.h>

#include<math.h>

#include<graphics.h>

void main()
{
int rcode_begin[4]={0,0,0,0},rcode_end[4]={0,0,0,0},region_code[4];
int W_xmax,W_ymax,W_xmin,W_ymin,flag=0;
float slope;
int x,y,x1,y1,i, xc,yc;

int gd=DETECT,gm;
initgraph(&gd,&gm,"C:\\\\TURBOC3\\\\BGI");

printf("\n***** Cohen Sutherland Line Clipping algorithm
*****");

printf("\n Now, enter XMin, YMin =");
scanf("%d %d",&W_xmin,&W_ymin);
printf("\n First enter XMax, YMax =");
scanf("%d %d",&W_xmax,&W_ymax);
printf("\n Please enter intial point x and y= ");
scanf("%d %d",&x,&y);
printf("\n Now, enter final point x1 and y1= ");
scanf("%d %d",&x1,&y1);
cleardevice();
//before clipping
```

```

rectangle(W_xmin,W_ymin,W_xmax,W_ymax);
line(x,y,x1,y1);

if(y>W_ymax)
{
    rcode_begin[0]=1;        // Top
    flag=1 ;
}
if(y<W_ymin)
{
    rcode_begin[1]=1;        // Bottom
    flag=1;
}
if(x>W_xmax)
{
    rcode_begin[2]=1;        // Right
    flag=1;
}
if(x<W_xmin)
{
    rcode_begin[3]=1;        //Left
    flag=1;
}

//end point of Line
if(y1>W_ymax)
{
    rcode_end[0]=1;          // Top
    flag=1;
}
if(y1<W_ymin)
{

```

```

        rcode_end[1]=1;           // Bottom
        flag=1;
    }
    if(x1>W_xmax)
    {
        rcode_end[2]=1;           // Right
        flag=1;
    }
    if(x1<W_xmin)
    {
        rcode_end[3]=1;           //Left
        flag=1;
    }
    if(flag==0)
    {
        printf("No need of clipping as it is already in window");
    }
    flag=1;
    for(i=0;i<4;i++)
    {
        region_code[i]= rcode_begin[i] && rcode_end[i] ;
        if(region_code[i]==1)
            flag=0;
    }
    if(flag==0)
    {
        printf("\n Line is completely outside the window");
    }
    else
    {
        slope=(float) (y1-y) / (x1-x) ;
        if(rcode_begin[2]==0 && rcode_begin[3]==1)    //left

```

```

{
    y=y+(float) (W_xmin-x)*slope ;
    x=W_xmin;
}
if(rcode_begin[2]==1 && rcode_begin[3]==0)        // right
{
    y=y+(float) (W_xmax-x)*slope ;
    x=W_xmax;
}
if(rcode_begin[0]==1 && rcode_begin[1]==0)        // top
{
    x=x+(float) (W_ymax-y)/slope ;
    y=W_ymax;
}
if(rcode_begin[0]==0 && rcode_begin[1]==1)        // bottom
{
    x=x+(float) (W_ymin-y)/slope ;
    y=W_ymin;
}
// end points
if(rcode_end[2]==0 && rcode_end[3]==1)    //left
{
    y1=y1+(float) (W_xmin-x1)*slope ;
    x1=W_xmin;
}
if(rcode_end[2]==1 && rcode_end[3]==0)        // right
{
    y1=y1+(float) (W_xmax-x1)*slope ;
    x1=W_xmax;
}
if(rcode_end[0]==1 && rcode_end[1]==0)        // top
{

```

```

        x1=x1+(float) (W_ymax-y1)/slope ;
        y1=W_ymax;
    }
    if(rcode_end[0]==0 && rcode_end[1]==1)        // bottom
    {
        x1=x1+(float) (W_ymin-y1)/slope ;
        y1=W_ymin;
    }
}
outtext("Before Clipping");
getch();
clearviewport();
//after clipping
outtext("After clipping");
rectangle(W_xmin,W_ymin,W_xmax,W_ymax);
setcolor(RED);
line(x,y,x1,y1);
getch();
closegraph();
}

```

Screenshots :

```

***** Cohen Sutherland Line Clipping algorithm *****
Now, enter XMin, YMin =100 100

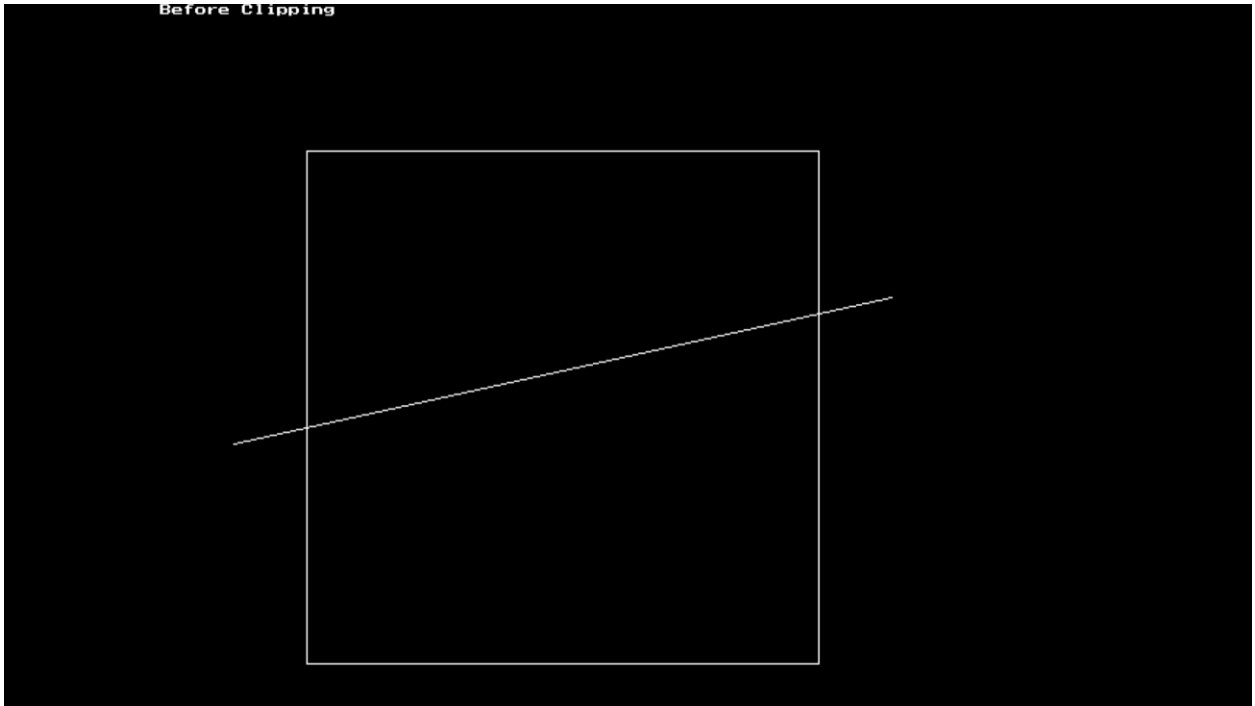
First enter XMax, YMax =450 450

Please enter intial point x and y= 500 200

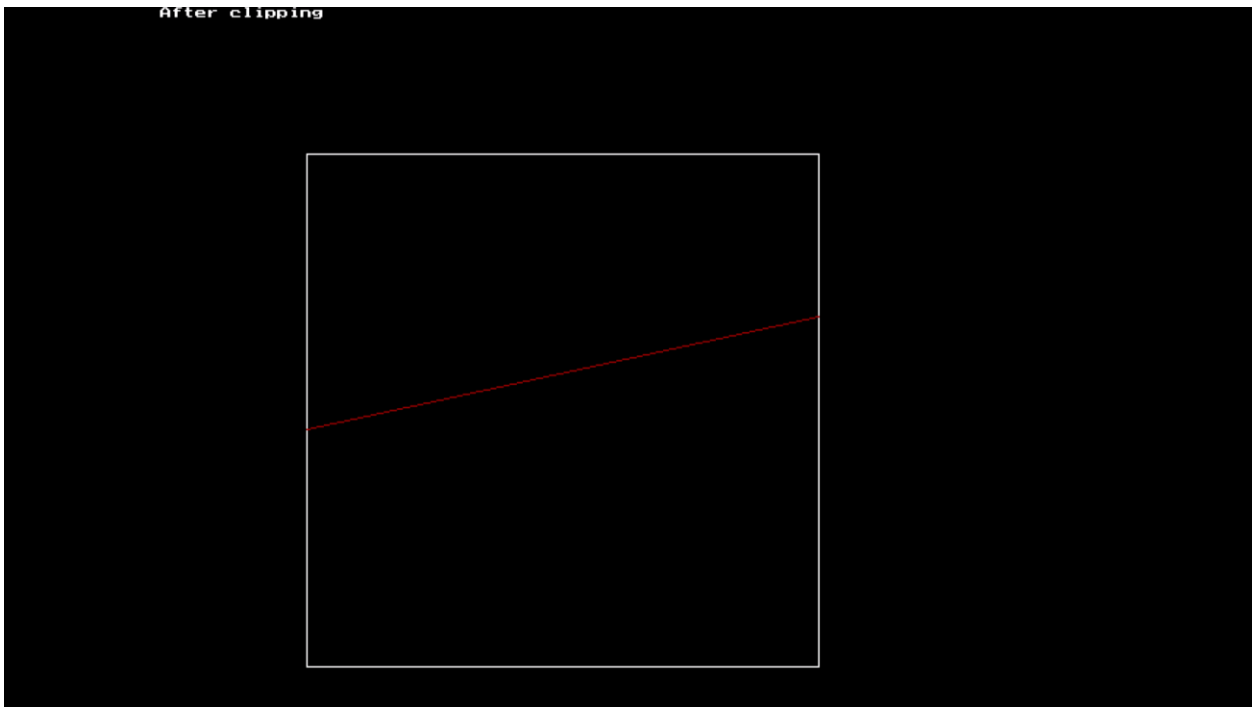
Now, enter final point x1 and y1= 50 300

```

Before Clipping



After clipping



Task 14

Demonstrate rotation of any shape.

```
// This a program to rotate a line
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
#include<math.h>
#include<stdlib.h>
#define PI 3.1415

void rotate(int, int, int, int);

void main()
{
    int gd=DETECT,gm;
    int x1=200,y1=200,x2=400,y2=200;

    initgraph(&gd,&gm,"c:\\turboc3\\bgi");
    outtext("This is a program for Rotation.");
    outtextxy(50,30,"Press any key to continue");
    line(x1,y1,x2,y2); //original shape
    getch();
    clearviewport();
    closegraph();

    rotate(x1,y1,x2,y2); //calling the function for rotation

    getch();
    closegraph();
}
```

```

}

void rotate(int x1, int y1, int x2, int y2)
{
    int gd=DETECT, gm;
    float r, c, s, angle;
    r=PI/180; //for converting degree to radian

    printf("Enter the angle to be rotated :");
    scanf("%f", &angle);
    //printf("%f", angle);
    //getch();

    initgraph(&gd, &gm, "c:\\turbo3\\bgi");
    outtext("Rotation : ");
    line(x1, y1, x2, y2); //original shape

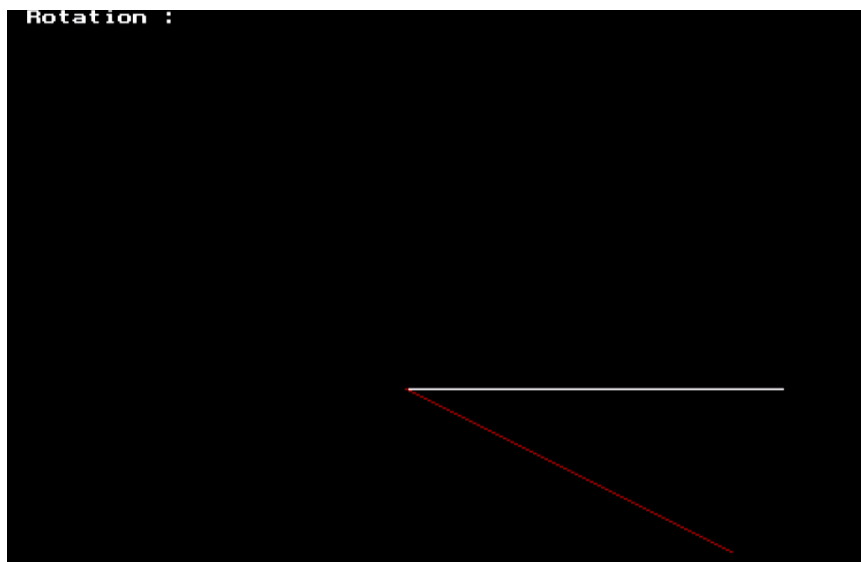
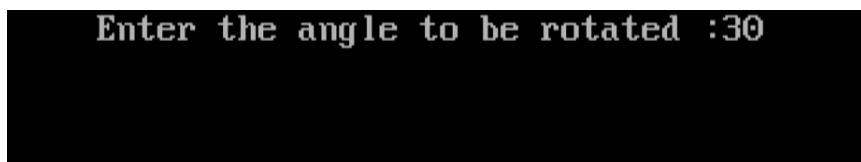
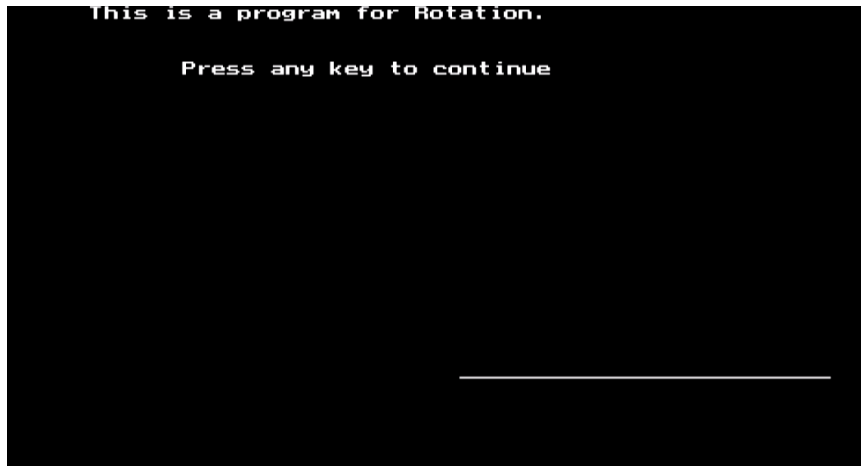
    c=cos(angle*r); //cos(radian)
    s=sin(angle*r); //sin(radian)

    //rotating the coordinates.
    x2=(int) (x1+((x2-x1)*c-(y2-y1)*s));
    y2=(int) (y1+((x2-x1)*s+(y2-y1)*c));

    setcolor(RED); //changing the color to red
    line(x1, y1, x2, y2); //rotated shape
}

```


Screenshots :



Task 15

Demonstrate translation of any shape.

```
// This a program to translate an object
#include<stdio.h>
#include<conio.h>
#include<graphics.h>

void  x_translate(int, int, int, int, int);
void  y_translate(int, int, int, int, int);
void  translate(int, int, int, int, int);

void main()
{
    int gd=DETECT,gm;
    int x1=50,y1=50,x2=150,y2=100,T,choice;

    initgraph(&gd,&gm,"c:\\turbo3\\bgi");
    outtext("This is a program for translation.");
    outtextxy(50,30,"Press any key to continue");
    rectangle(x1,y1,x2,y2); //original shape
    getch();
    closegraph();

    printf("Enter Your Choice :");
    printf("\n1. Translate Horizontally");
    printf("\n2. Translate Vertically");
    printf("\n3. Translate Diagonally]\n\n");
    scanf("%d",&choice);
    printf("\n\nEnter the translation factor :  ");
    scanf("%d",&T);
```

```

switch(choice)
//calling the functions according to the choice
{
    case 1: x_translate(x1,y1,x2,y2,T);
    break;
    case 2: y_translate(x1,y1,x2,y2,T);
    break;
    case 3: translate(x1,y1,x2,y2,T);
    break;
    default: printf("\nInvalid Choice!");
}
getch();
closegraph();
}

void x_translate(int x1, int y1, int x2, int y2, int T)
{
    int gd=DETECT,gm;

    initgraph(&gd,&gm,"c:\\turbo3\\bgi");
    rectangle(x1,y1,x2,y2);//original shape
    x1=x1+T;                //horizontal translation
    x2=x2+T;
    setcolor(RED);          //changing the color to red
    rectangle(x1,y1,x2,y2);//translated shape
}

void y_translate(int x1, int y1, int x2, int y2, int T)
{
    int gd=DETECT,gm;

```

```

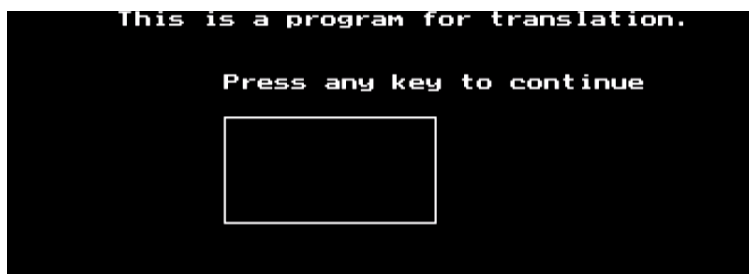
initgraph(&gd,&gm,"c:\\turbo3\\bgi");
rectangle(x1,y1,x2,y2); //original shape
y1=y1+T;                //vertical translation
y2=y2+T;
setcolor(RED);          //changing the colour to red
rectangle(x1,y1,x2,y2); //translated shape
}

void translate(int x1, int y1, int x2, int y2, int T)
{
    int gd=DETECT,gm;

    initgraph(&gd,&gm,"c:\\turbo3\\bgi");
    rectangle(x1,y1,x2,y2); //original shape
    x1=x1+T;                //horizontal
    x2=x2+T;                //translation
    y1=y1+T;                //vertical
    y2=y2+T;                //translation
    setcolor(RED);          //changing the colour to red
    rectangle(x1,y1,x2,y2); //translated shape
}

```

Screenshots :



Enter Your Choice :
1. Translate Horizontally
2. Translate Vertically
3. Translate Diagonally]

1

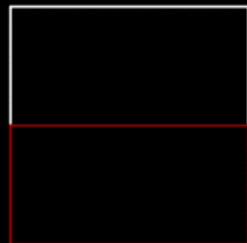
Enter the translation factor : 50_



Enter Your Choice :
1. Translate Horizontally
2. Translate Vertically
3. Translate Diagonally]

2

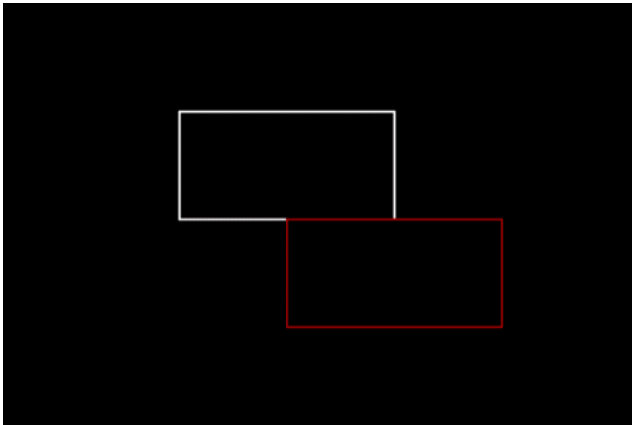
Enter the translation factor : 50_



Enter Your Choice :
1. Translate Horizontally
2. Translate Vertically
3. Translate Diagonally

3

Enter the translation factor : 50_



Task 16

Develop any 2D Game of your choice

```
/* This is a 2D car racing game named "Drag 'N' Drive." */

#include<graphics.h>
#include<stdio.h>
#include<conio.h>
#include<time.h>    // for rand() function.
#include<stdlib.h>  // for rand() function.
#include<dos.h>     // for delay() and exit() function.
#define N 10       // Number of high scores
#define FNAME "myscore.txt" // for a file named myscore.txt in bin
folder.

int a=0; // for score
char scr[50]; // for printing score using printf.
int cars=0; // for car selection
int life=3; //provides 3 chances to players.
int D=50; //delay
int score[N];
int count=0;
FILE *fp; // file pointer
int myscore;

void loading();
void car(int);
void track(int);
void play();
void menu();
void high_score();
```

```

void instructions();
void about();
void text();
void text1();
void heart1();
void heart2();
void heart3();
void speed();
void info();
void h1();
void h2();
void select_car();
void car_choice();
void car1();
void car2();
void car3();
void car4();
void mycar1(int);
void mycar2(int);
void mycar3(int);
void mycar4(int);

int main()
{
    int gd=DETECT, gm;
    initgraph(&gd, &gm, "c:\\turbo3\\bgi");
    loading(); //loading screen
    menu();
    return 0;
}

void play()

```



```

{
    int i,j,k,l,r,s,ch;
    cleardevice();
    setbkcolor(8);

    r=210+rand()%150;
    // generates a random number between 210 and 210+150
    s=210+rand()%150;

    for(k=0;k<433;k+=1)
    {
        for(l=-200;l<433;l+=1)
        {
            for(i=205;i<345;i++)
            {
                for(j=0;j<500;j++)
                {
                    track(k);
                    speed();
                    info();
                    switch(life)
                //to show the hearts corressponding to remaining life.
                {
                    case 1: heart1();
                    break;
                    case 2: heart2();
                    break;
                    case 3: heart3();
                    break;
                }
                switch(cars)
                {

```

```

        case 1 : mycar1(i);
        break;
        case 2 : mycar2(i);
        break;
        case 3 : mycar3(i);
        break;
        case 4 : mycar4(i);
        break;
        default : outtextxy(200,200,"No car selected.");
    }

    rectangle(r,k,r+50,50+k);
    rectangle(s,l,s+50,50+l);

    if(i+50>=r&&i<=(r+40)&&k>=265&&k<=415)
    {
        cleardevice();
        if(life!=1)
        {
            settextstyle(1,0,3);
            outtextxy(200,getmaxy()/2,"One Chance Lost!");
            circle(420,250,15);
            circle(426,246,1);
            circle(414,246,1);
            arc(420,259,10,160,5);
            life=life-1;    //Reduces life.
            outtextxy(200,180,"Current Score :");
            outtextxy(380,180,scr);
            outtextxy(190,300,"Press Enter to Continue!");
            kbhit();

            while(getch()!=13)

//to hold the screen till enter is pressed

```

```

        {
            getch();
        }
play();
}
else if(life==1)
{
    settextstyle(1,0,6);
    outtextxy(160,160,"GAME OVER");
    settextstyle(1,0,3);
    outtextxy(180,getmaxy()/2,"Your Score: ");
    outtextxy(340,getmaxy()/2,scr);
    score[count]=a;
    count=count+1;
    fp=fopen(FNAME,"wb");
    if(a>myscore)
    {
        myscore=a;
        putw(myscore,fp);
    }
    else
    {
        myscore=myscore;
        putw(myscore,fp);
    }
    fclose(fp);
    a=0;        //resets score to 0.
    life=3;
    D=50;       //delay resets to 50
    outtextxy(120,360,"Press Enter to return to
the main Menu.");
    kbhit();

```

```

        while(getch() !=13)
//to hold the screen till enter is pressed
        {
            getch();
        }
        menu();
    }
}
if(i+50>=s&&i<=(s+40) &&l>=265&&l<=415)
{
    cleardevice();
    if(life!=1)
    {
        settextstyle(1,0,3);
        outtextxy(200,getmaxy()/2,"One Chance Lost!");
        circle(420,250,15);
        circle(426,246,1);
        circle(414,246,1);
        arc(420,259,10,160,5);

        life=life-1;    //Reduces life.
        outtextxy(200,150,"Current Score: ");
        outtextxy(380,150,scr);
        outtextxy(200,350,"Press Enter to Continue!");
        kbhit();
        while(getch() !=13)
//to hold the screen till enter is pressed
        {
            getch();
        }
        play();
    }
}

```

```

else if(life==1)
{
    settextstyle(1,0,6);
    outtextxy(160,160,"GAME OVER");
    settextstyle(1,0,3);
    outtextxy(220,230,"Your Score :");
    outtextxy(300,150,scr);
    score[count]=a;
    count=count+1;
    fp=fopen(FNAME,"wb");
    if(a>myscore)
    {
        myscore=a;
        putw(myscore,fp);
    }
    else
    {
        myscore=myscore;
        putw(myscore,fp);
    }
    fclose(fp);
    a=0;      //resets score to 0.
    life=3;   //resets life to 3.
    D=50;     //delay resets to 50.
    outtextxy(120,360,"Press Enter to return to
the main Menu.");

    kbhit();

    while(getch()!=13) //to hold the screen
    {
        getch();
    }
}
till enter is pressed

```

```

        menu() ;
    }
}

if(k==430)
{
    k=0;
    a=a+10;
    if(a>=50&&a<100)
    {
        D=45;
    }
    else if (a>=100&&a<200)
    {
        D=40;
    }
    else if(a>=200&&a<300)
    {
        D=35;
    }
    else if(a>=300&&a<400)
    {
        D=30;
    }
    else if(a>=400&&a<600)
    {
        D=25;
    }
    else if(a>=600)
    {
        D=20;
    }
}

```

```

        r=210+rand()%150;
// generates a random number between 210 and 210+150
    }
    else if(l==430)
    {
        l=0;
        a=a+10;
        if(a>=50&&a<100)
        {
            D=45;
        }
        else if (a>=100&&a<200)
        {
            D=40;
        }
        else if(a>=200&&a<300)
        {
            D=35;
        }
        else if(a>=300&&a<400)
        {
            D=30;
        }
        else if(a>=400&&a<600)
        {
            D=25;
        }
        else if(a>=600)
        {
            D=20;
        }
        s=210+rand()%150;

```

```

    }
    else
    {
        k+=10;
        l+=10;
    }

    if(kbhit())
    {
        ch=getch();
        sound(400);
        delay(10);
        if(ch==32) //space button
        {
            nosound();
            settextstyle(1,0,3);
            outtextxy(250,200,"GAME PAUSED");
            outtextxy(250,150,"Current Score :");
            outtextxy(450,150,scr);
            outtextxy(200,250,"Press Space bar to
Resume");

            while(getch()!=32)
                //to hold the screen till Space is pressed
            {
                getch();
            }
        }
        if(ch==75&& i>214) //move left
        {
            if(i==230)
            {
                i=250;

```



```

        }
        i-=20;
    }

    if(ch==77&& i<331) //move right
    {
        if(i==330)
//max value of x coordinate of car
        {
            i=310; // to remain on the track
        }
        i+=20;
    }

    if(ch==27) //escape key is pressed
    {
        nosound();
        menu();
    }
    nosound();
}
delay(D);
cleardevice();
}
}
getch();
}

void select_car()
{

```

```

int x1=95,y1=50,x2=195,y2=391,z=120,c=0;
cleardevice();
setbkcolor(0);
while(1)
{
car_choice();
car1();
car2();
car3();
car4();
rectangle(x1,y1,x2,y2); //cursor

    if(kbhit())
    {
        c=getch();
        if(c==27) //esc is pressed
        {
            menu();
        }
        else if(c==75&&x1==95) //left arrow key and cursor at
the left most box
        {
            cleardevice();
            car_choice();
            car1();
            car2();
            car3();
            car4();
            x1=455;
            x2=555;

            // setcolor(RED);
            rectangle(x1,y1,x2,y2);

```

```

    }
    else if(c==75) //left arrow key
    {
        cleardevice();
        car_choice();
        car1();
        car2();
        car3();
        car4();
        x1=x1-z;
        x2=x2-z;

        // setcolor(RED);
        rectangle(x1,y1,x2,y2);
    }
    else if(c==77&&x1==455)
//right arrow key and the cursor at the rightmost box
    {
        cleardevice();
        car_choice();
        car1();
        car2();
        car3();
        car4();
        x1=95;
        x2=195;

        //setcolor(RED);
        rectangle(x1,y1,x2,y2);
    }
    else if(c==77) //right arrow key
    {
        cleardevice();
        car_choice();

```

```

        car1();
        car2();
        car3();
        car4();
        x1=x1+z;
        x2=x2+z;
        //setcolor(RED);
        rectangle(x1,y1,x2,y2);
    }

else if(c==13)    // if Enter key is pressed.
{
    if(x1==95)    //position of the cursor
    {
        cars=3;
        play();  //car to be selected
    }
    else if(x1==215)
    {
        cars=1;
        play();
    }
    else if(x1==335)
    {
        cars=2;
        play();
    }
    else if(x1==455)
    {
        cars=4;
        play();
    }
}

```

```

        }

    }

}

void car_choice()
{
    int i=120;
    bar3d(325-i-i,45,325+i+i,45,2,1);
    bar3d(325-i-i,396,325+i+i,396,2,1);

    bar3d(325,45,327,396,2,1);
    bar3d(325-i,45,327-i,396,2,1);
    bar3d(325-i-i,45,327-i-i,396,2,1);
    bar3d(325+i,45,327+i,396,2,1);
    bar3d(325+i+i,45,327+i+i,396,2,1);

    settextstyle(7,0,5); //font_direction_fontsize
    outtextxy(265-i,80,"1");
    outtextxy(265,80,"2");
    outtextxy(265+i,80,"3");
    outtextxy(265+i+i,80,"4");

    setcolor(GREEN);
    settextstyle(7,0,4);
    outtextxy(200,410,"Select your car");
    settextstyle(1,0,2);
    outtextxy(180,450,"Use arrow keys to navigate.");
}

void car1()
{

```

```

setcolor(RED);
rectangle(250,250,280,290); //roof
rectangle(245,290,285,300);
setcolor(GREEN);
rectangle(235,230,295,320); //main outline of car
arc(265,250,10,170,35); //side mirror's
arc(265,190,10,170,15); //front light

line(235,230,250,250); //top_right_corner
line(295,230,280,250); //top_left_corner
line(245,300,235,320); //bottom_left_corner
line(285,300,295,320); //bottom_right_corner

setcolor(GREEN);
ellipse(265,320,180,360,30,20); //trunk lid
ellipse(265,230,0,180,30,55); //bonet
}

void mycar1(int i)
{
    setcolor(RED);
    rectangle(20+i,350,40+i,380); //roof
    rectangle(15+i,380,45+i,385);
    setcolor(GREEN);
    rectangle(10+i,340,50+i,395); //main outline of car
    arc(30+i,355,10,170,24); //side mirror's
    arc(30+i,320,10,170,10); //front light

    line(50+i,340,40+i,350); //top_right_corner
    line(10+i,340,20+i,350); //top_left_corner
    line(15+i,385,10+i,395); //bottom_left_corner
    line(45+i,385,50+i,395); //bottom_right_corner

```

```

        setcolor(GREEN);
        ellipse(30+i,395,180,360,20,10); //trunk lid
        ellipse(30+i,340,0,180,20,30); //bonet
    }

void car2()
{
    int i=120;
    setcolor(BLUE);
    rectangle(250+i,250,280+i,290); //roof
    rectangle(245+i,290,285+i,300);
    setcolor(YELLOW);
    rectangle(235+i,230,295+i,320); //main outline of car
    arc(265+i,250,10,170,35); //side mirror's
    arc(265+i,190,10,170,15); //front light

    line(235+i,230,250+i,250); //top_right_corner
    line(295+i,230,280+i,250); //top_left_corner
    line(245+i,300,235+i,320); //bottom_left_corner
    line(285+i,300,295+i,320); //bottom_right_corner

    setcolor(YELLOW);
    ellipse(265+i,320,180,360,30,20); //trunk lid
    ellipse(265+i,230,0,180,30,55); //bonet
}

void mycar2(int i)
{
    setcolor(BLUE);
    rectangle(20+i,350,40+i,380); //roof
    rectangle(15+i,380,45+i,385);

```

```

setcolor(YELLOW);
rectangle(10+i,340,50+i,395);      //main outline of car
arc(30+i,355,10,170,24);           //side mirror's
arc(30+i,320,10,170,10);           //front light

line(50+i,340,40+i,350); //top_right_corner
line(10+i,340,20+i,350); //top_left_corner
line(15+i,385,10+i,395); //bottom_left_corner
line(45+i,385,50+i,395); //bottom_right_corner

setcolor(YELLOW);
ellipse(30+i,395,180,360,20,10); //trunk lid
ellipse(30+i,340,0,180,20,30); //bonet
}

void car3()
{
    int i=120;
    setcolor(RED);
    rectangle(250-i,250,280-i,290); //roof
    setcolor(RED);
    rectangle(235-i,230,295-i,320);      //main outline of car
    arc(265-i,250,10,170,35);           //side mirror's
    arc(265-i,190,10,170,15);           //front light

    line(235-i,230,250-i,250); //top_right_corner
    line(295-i,230,280-i,250); //top_left_corner
    line(250-i,290,235-i,320); //bottom_left_corner
    line(280-i,290,295-i,320); //bottom_right_corner

    setcolor(RED);
    ellipse(265-i,320,180,360,30,20); //trunk lid

```



```

        ellipse(265-i,230,0,180,30,55); //bonet
    }
void mycar3(int i)
{
    setcolor(RED);
    rectangle(20+i,350,40+i,380); //roof
    setcolor(RED);
    rectangle(10+i,340,50+i,395); //main outline of car
    arc(30+i,355,10,170,24); //side mirror's
    arc(30+i,320,10,170,10); //front light

    line(50+i,340,40+i,350); //top_right_corner
    line(10+i,340,20+i,350); //top_left_corner
    line(20+i,380,10+i,395); //bottom_left_corner
    line(40+i,380,50+i,395); //bottom_right_corner

    setcolor(RED);
    ellipse(30+i,395,180,360,20,10); //trunk lid
    ellipse(30+i,340,0,180,20,30); //bonnet
}

void car4()
{
    int i=120;
    setcolor(GREEN);
    rectangle(250+i+i,250,280+i+i,290); //roof
    rectangle(245+i+i,290,285+i+i,300);
    setcolor(WHITE);
    rectangle(235+i+i,230,295+i+i,320); //main outline of car
    arc(265+i+i,250,10,170,35); //side mirror's
    arc(265+i+i,190,10,170,15); //front light

```

```

    line(235+i+i,230,250+i+i,250); //top_right_corner
    line(295+i+i,230,280+i+i,250); //top_left_corner
    line(245+i+i,300,235+i+i,320); //bottom_left_corner
    line(285+i+i,300,295+i+i,320); //bottom_right_corner

    setcolor(WHITE);
    ellipse(265+i+i,320,180,360,30,20); //trunk lid
    ellipse(265+i+i,230,0,180,30,55); //bonnet
}

void mycar4(int i)
{
    setcolor(GREEN);
    rectangle(20+i,350,40+i,380); //roof
    rectangle(15+i,380,45+i,385);
    setcolor(WHITE);
    rectangle(10+i,340,50+i,395); //main outline of car
    arc(30+i,355,10,170,24); //side mirror's
    arc(30+i,320,10,170,10); //front light

    line(50+i,340,40+i,350); //top_right_corner
    line(10+i,340,20+i,350); //top_left_corner
    line(15+i,385,10+i,395); //bottom_left_corner
    line(45+i,385,50+i,395); //bottom_right_corner

    setcolor(WHITE);
    ellipse(30+i,395,180,360,20,10); //trunk lid
    ellipse(30+i,340,0,180,20,30); //bonnet
}

void track(int k)
{
    setcolor(WHITE);

```

```

bar3d(120,0,125,478,5,3); //left boundary
bar3d(480,0,485,478,5,3); //right boundary
line(280,0,280,getmaxy()); //lanes
line(340,0,340,getmaxy());

    // for moving tracks
    setcolor(WHITE);
    bar3d(200,-420+k,205,-380+k,5,3);
    bar3d(410,-420+k,405,-380+k,5,3);
    bar3d(200,-360+k,205,-320+k,5,3);
    bar3d(410,-360+k,405,-320+k,5,3);
    bar3d(200,-300+k,205,-260+k,5,3);
    bar3d(410,-300+k,405,-260+k,5,3);
    bar3d(200,-240+k,205,-200+k,5,3);
    bar3d(410,-240+k,405,-200+k,5,3);
    bar3d(200,-180+k,205,-140+k,5,3);
    bar3d(410,-180+k,405,-140+k,5,3);
    bar3d(200,-120+k,205,-80+k,5,3);
    bar3d(410,-120+k,405,-80+k,5,3);
    bar3d(200,-60+k,205,-20+k,5,3);
    bar3d(410,-60+k,405,-20+k,5,3);
    bar3d(200,0+k,205,40+k,5,3);
    bar3d(410,0+k,405,40+k,5,3);
    bar3d(200,60+k,205,100+k,5,3);
    bar3d(410,60+k,405,100+k,5,3);
    bar3d(200,120+k,205,160+k,5,3);
    bar3d(410,120+k,405,160+k,5,3);
    bar3d(200,180+k,205,220+k,5,3);
    bar3d(410,180+k,405,220+k,5,3);
    bar3d(200,240+k,205,280+k,5,3);
    bar3d(410,240+k,405,280+k,5,3);
    bar3d(200,300+k,205,340+k,5,3);

```

```

        bar3d(410,300+k,405,340+k,5,3);
        bar3d(200,360+k,205,400+k,5,3);
        bar3d(410,360+k,405,400+k,5,3);
        bar3d(200,420+k,205,460+k,5,3);
        bar3d(410,420+k,405,460+k,5,3);
    }

void loading() //loading screen
{
    int i,j;
    for(j=0;j<20;j++) // For Blinking of text
    {
        settextstyle(4,0,8);
        setcolor(j);
        outtextxy(70,100,"Drag 'N' Drive");
        delay(100);
    }
    settextstyle(6,0,2);
    outtextxy((getmaxx()/2)-30,getmaxy()-130,"LOADING");
    for(i=0;i<400;i+=2) //loading animation
    {
        setcolor(RED);
        rectangle((getmaxx()/2)-200+i,getmaxy()-
100,(getmaxx()/2)+200,getmaxy()-80);
        delay(10);
    }
}

void menu()
{
    int i,k,x1=210,y1=200,x2=440,y2=247;
    cleardevice();
    setbkcolor(1);

```

```

while(1)
{
    h1();
    h2();
    setcolor(RED);
    rectangle(x1,y1,x2,y2);    //cursor

if(kbhit()) // detects the pressing of key
{
    k=getch();
    if(k==72&&y1==200) //cursor at the top
    {
        y1=200+160;    //sends the cursor to bottom
        y2=247+160;
        cleardevice();
        h1();
        h2();
        setcolor(RED);
        rectangle(x1,y1,x2,y2);
    }
    else if(k==80&&y2==407)
//cursor at the bottom and moves to the top
    {
        x1=210; //resets the cursor position to top
        y1=200;
        x2=440;
        y2=247;
        cleardevice();
        h1();
        h2();
        setcolor(RED);
        rectangle(x1,y1,x2,y2);
    }
}
}

```

```

}
else if(k==80) //cursor moves downward
{
    cleardevice();
    h1();
    h2();
    y1=y1+40;
    y2=y2+40;
    rectangle(x1,y1,x2,y2);
}
else if(k==72) //cursor moves upward
{
    cleardevice();
    h1();
    h2();
    y1=y1-40;
    y2=y2-40;
    setcolor(RED);
    rectangle(x1,y1,x2,y2);
}

else if(k==13) //Enter key is pressed
{
    if(y1==200) // cursor on the first option
    {
        cleardevice();
        select_car(); // opens the game
    }
    else if(y1==240) //High Score
    {
        cleardevice();
        high_score();
    }
}

```

```

        }
        else if(y1==280)
        {
            cleardevice();
            instructions(); //Instructions
        }
        else if(y1==320)
        {
            cleardevice();
            about(); //About
        }
        else if(y1==360)
        {
            exit(0);
            closegraph();
        }
    }
}
}

void high_score()
{
    int p,q,tmp,score1,score2,score3,score4,score5;
    char as[100],ss[100]; //for sprintf
    setbkcolor(0); //Black
    fp=fopen(FNAME, "rb");
    for (p=0;p<N;p++)
    {
        for(q=p+1;q<N;q++)
        {
            if (score[p]<score[q])

```

```

        {
            tmp = score[p];
            score[p] = score[q];
            score[q] = tmp;
        }
    }

myscore=getw(fp);
score1=score[0];
score2=score[1];
score3=score[2];
score4=score[3];
score5=score[4];

sprintf(as," %d ",myscore);
fclose(fp);

sprintf(ss,"1. %d | 2. %d | 3. %d | 4. %d | 5. %d
",score1,score2,score3,score4,score5); // stores high score in buffer

settextstyle(4,0,7);
setcolor(GREEN);
outtextxy(150,20,"High Scores");
bar3d(50,120,550,125,3,1);
settextstyle(8,0,3);
setcolor(BLUE);
outtextxy(60,150," All time High Score :");
settextstyle(8,0,2);
setcolor(WHITE);
outtextxy(80,180,as); // displays the high score
settextstyle(8,0,3);
setcolor(BLUE);
outtextxy(60,260,"Current High Scores :");

```



```

    settextstyle(8,0,2);
    setcolor(WHITE);
    outtextxy(60,300,ss);
    setcolor(RED);
    bar3d(50,420,550,415,3,1);
    settextstyle(1,0,1);
    outtextxy(100,455,"Press Esc to return to the Main Menu");

    kbhit();
    while(getch()!=27)    //Esc key
    {
        getch();
    }
    cleardevice();
    menu();
}

void heart1()
{
    int
h1[]={517,75,506,60,510,50,515,51,517,60,520,51,525,50,529,60,517,75};
    setcolor(RED);
    setfillstyle(SOLID_FILL,RED);
    fillpoly(9,h1);
}

void heart2()
{
    int
h1[]={517,75,506,60,510,50,515,51,517,60,520,51,525,50,529,60,517,75};
    int
h2[]={557,75,546,60,550,50,555,51,557,60,560,51,565,50,569,60,557,75};
    setcolor(RED);
    setfillstyle(SOLID_FILL,RED);

```

```

        fillpoly(9,h1);
        fillpoly(9,h2);
    }
void heart3()
{
    int
h1[]={517,75,506,60,510,50,515,51,517,60,520,51,525,50,529,60,517,75};
    int
h2[]={557,75,546,60,550,50,555,51,557,60,560,51,565,50,569,60,557,75};
    int
h3[]={597,75,586,60,590,50,595,51,597,60,600,51,605,50,609,60,597,75};
    setcolor(RED);
    setfillstyle(SOLID_FILL,RED);
    fillpoly(9,h1);
    fillpoly(9,h2);
    fillpoly(9,h3);
}
void speed()
{
    if(D==50)
    {
        settextstyle(8,0,2);
        outtextxy(500,320,"Speed:");
        settextstyle(7,0,2);
        outtextxy(500,350,"100 mph");
    }
    else if(D==45)
    {
        settextstyle(8,0,2);
        outtextxy(500,320,"Speed:");
        settextstyle(7,0,2);
        outtextxy(500,350,"110 mph");
    }
}

```

```

else if (D==40)
{
    settextstyle(8,0,2);
    outtextxy(500,320,"Speed:");
    settextstyle(7,0,2);
    outtextxy(500,350,"120 mph");
}
else if (D==35)
{
    settextstyle(8,0,2);
    outtextxy(500,320,"Speed:");
    settextstyle(7,0,2);
    outtextxy(500,350,"130 mph");
}
else if (D==30)
{
    settextstyle(8,0,2);
    outtextxy(500,320,"Speed:");
    settextstyle(7,0,2);
    outtextxy(500,350,"140 mph");
}
else if (D==25)
{
    settextstyle(8,0,2);
    outtextxy(500,320,"Speed:");
    settextstyle(7,0,2);
    outtextxy(500,350,"150 mph");
}
else if (D==20)
{
    settextstyle(8,0,2);
    outtextxy(500,320,"Speed:");

```

```

        settextstyle(7,0,2);
        outtextxy(500,350,"160 mph");
    }

}

void info()
{
    bar3d(10,0,630,0,5,2);
    // for boundaries of playing screen
    bar3d(10,0,10,475,5,2);
    bar3d(10,475,630,475,5,2);
    bar3d(630,0,630,475,5,2);

    settextstyle(4,0,2);
    outtextxy(20,50,"Controls:");
    settextstyle(13,0,5);
    outtextxy(20,100,"Left arrow:");
    outtextxy(20,120,"Move left");
    outtextxy(20,150,"Right arrow:");
    outtextxy(20,170,"Move right");
    outtextxy(20,200,"Space bar:");
    outtextxy(20,220,"Pause/Resume");
    outtextxy(20,250,"Esc:");
    outtextxy(20,270,"Exit");

    //prints score to the screen.
    settextstyle(8,0,2);
    outtextxy(500,150,"Score:");
    sprintf(scr,"%d ",a);
    settextstyle(7,0,2);
    outtextxy(500,180,scr);
    settextstyle(13,0,3);

```

```

        outtextxy(500,30,"Remaining Life:");
    }

void h1() //prints the "MENU"
{
    settextstyle(7,0,8);
    setcolor(GREEN);
    outtextxy(215,50,"MENU");
    setcolor(RED);
    bar3d(20,145,620,150,5,2);
}

void h2() // prints options
{
    settextstyle(3,0,5);
    setcolor(GREEN);
    outtextxy(220,195,"Play Game");
    outtextxy(220,235,"High Score");
    outtextxy(220,275,"Instructions");
    outtextxy(220,315,"About");
    outtextxy(220,355,"Exit");
}

void instructions()
{
    setbkcolor(BLACK);
    settextstyle(4,0,7);
    setcolor(GREEN);
    outtextxy(150,20,"Instructions");
    bar3d(50,120,550,125,3,1);
    settextstyle(4,0,4);
    setcolor(BLUE);
    outtextxy(60,135,"Objective:");
}

```

```

    settextstyle(8,0,3);
    setcolor(GREEN);
    outtextxy(60,165,"Save your car from the obstacles.");
    bar(50,200,550,202);
    settextstyle(4,0,4);
    setcolor(BLUE);
    outtextxy(60,203,"Controls:");
    settextstyle(8,0,3);
    setcolor(GREEN);
    outtextxy(60,235,"Left arrow key : Move left");
    outtextxy(60,265,"Right arrow key: Move right");
    outtextxy(60,295,"Up arrow key   : Move Up");
    outtextxy(60,325,"Down arrow key : Move Down");
    outtextxy(60,355,"space bar       : Pause/Resume game");
    outtextxy(60,385,"Enter          : Select/Choose/Continue");
    outtextxy(60,415,"Esc            : return to menu");

    bar3d(50,448,550,452,3,1);
    settextstyle(1,0,1);
    setcolor(RED);
    outtextxy(100,455,"Press esc to return to the main menu");
    kbhit();
    while(getch() !=27)
    {
        getch();
    }
    cleardevice();
    menu();
}

void about()
{

```

```

    setbkcolor (BLACK) ;
    settextstyle (4,0,7) ;
    setcolor (GREEN) ;
    outtextxy (150,20,"About") ;
    bar3d (50,120,550,125,3,1) ;
    setcolor (GREEN) ;
    bar3d (50,460,550,465,3,1) ;
    text () ;
    getch () ;
    cleardevice () ;
    text1 () ;
    settextstyle (4,0,7) ;
    setcolor (GREEN) ;
    outtextxy (150,20,"About") ;
    bar3d (50,120,550,125,3,1) ;
    settextstyle (1,0,1) ;
    setcolor (RED) ;
    outtextxy (100,455,"Press esc to return to the main menu") ;

    kbhit () ;
    while (getch () !=27)
    {
        getch () ;
    }
    cleardevice () ;
    menu () ;
}

void text ()
{
    settextstyle (4,0,4) ;
    setcolor (BLUE) ;

```

```

        outtextxy(40,130,"The Game: ");
        settextstyle(8,0,1);
        setcolor(GREEN);
        outtextxy(40,170,"Hope, You have played this game and enjoyed
it.");
        outtextxy(40,190,"If not, then try it and score as much as you
can.");
        outtextxy(40,220,"Drag'N'drive, as the name suggests, you will
have to");
        outtextxy(40,240,"drag your car away from the obstacles in order
to");
        outtextxy(40,260,"drive further. You have to select one of the
four");
        outtextxy(40,280,"given cars and drive it to score points by
avoiding");
        outtextxy(40,300,"the obstacles.");
        outtextxy(40,330,"Each obstacle avoided will award you 10 points.
You ");
        outtextxy(40,350,"also have 3 lives, which means you have 3
chances.");
        outtextxy(40,380,"You can also give us your valuable suggestions
via");
        outtextxy(40,400,"the details provided in the next page.");
        setcolor(WHITE);
        settextstyle(13,0,5);
        outtextxy(450,440,"Version 1.14");
    }
void text1()
{
    settextstyle(4,0,4);
    setcolor(BLUE);
    outtextxy(40,130,"The Developers: ");
    settextstyle(8,0,1);
    outtextxy(40,320,"Aniket.kumar@hotmail.com");
    outtextxy(40,350,"rupalisingh2709@gmail.com");
}

```



```

        outtextxy(400,380,"bit.ly/dragndrive");

        setcolor(GREEN);

        outtextxy(40,170,"This game is a part of Computer Graphics
Project work.");

        outtextxy(40,190,"We have tried to reduce as much technical or
logical ");

        outtextxy(40,210,"errors as we can. However, it may contain some
bugs.");

        outtextxy(40,240,"It is developed by Rupali Singh and Aniket
Kumar,");

        outtextxy(40,260,"students of BCA (DSVV,Haridwar).");

        outtextxy(40,290,"You can mail us your valuable suggestions at:
");

        outtextxy(40,380,"The repository of this game is at: ");

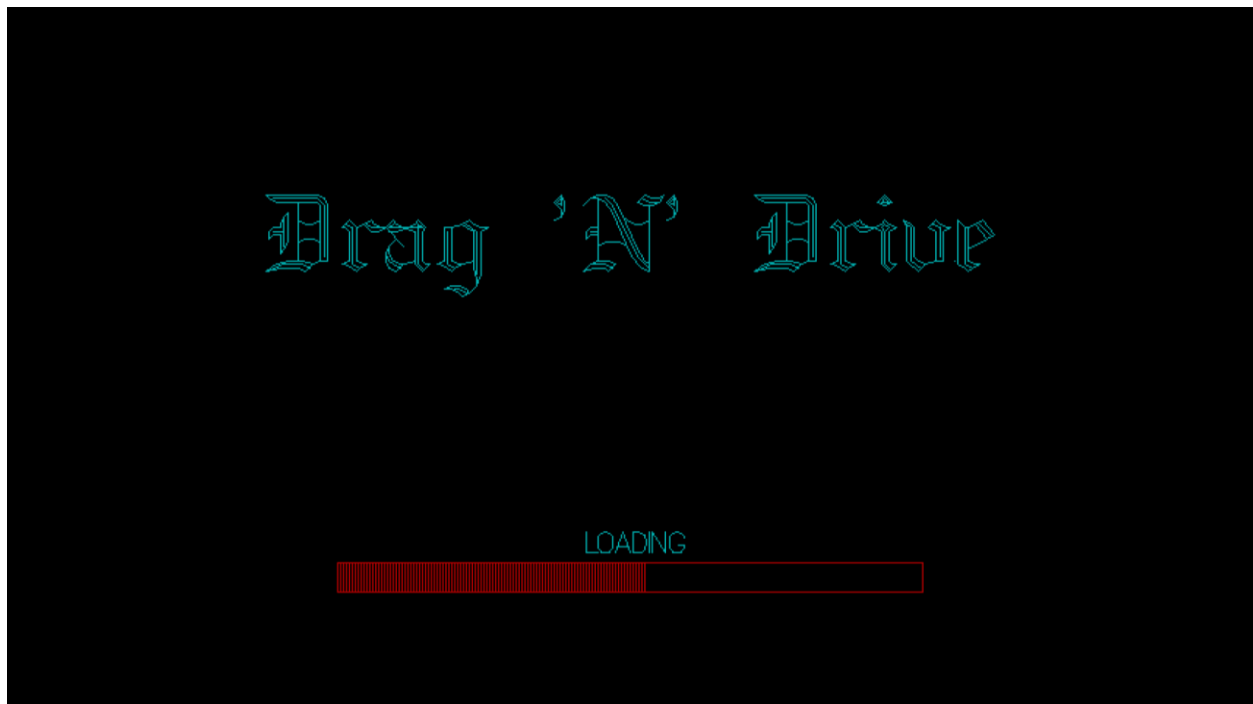
        outtextxy(40,410,"So, what are you waiting for?");

        outtextxy(40,430,"Go & score as much as you can.");

    }

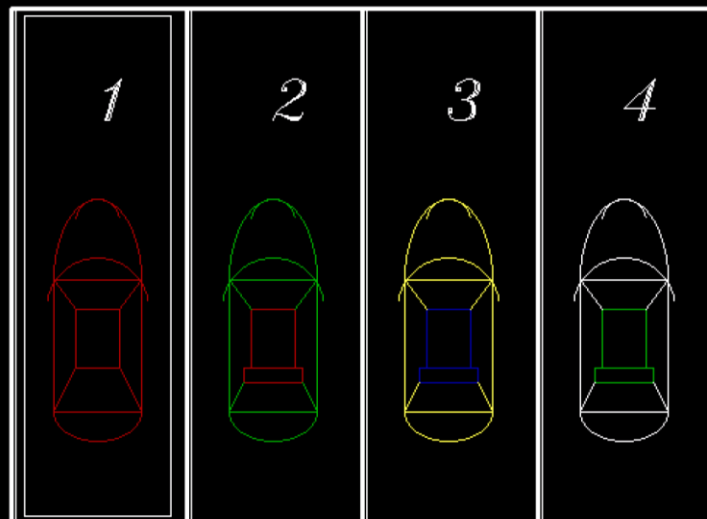
```

Screenshots :

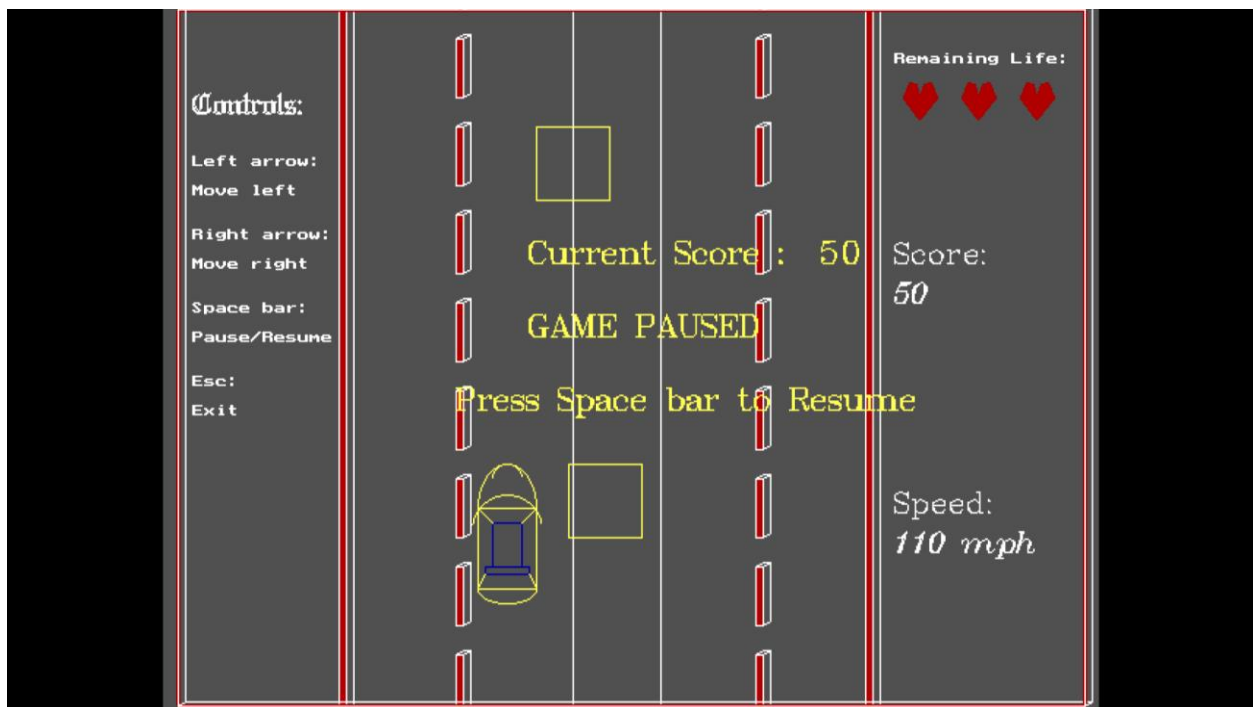
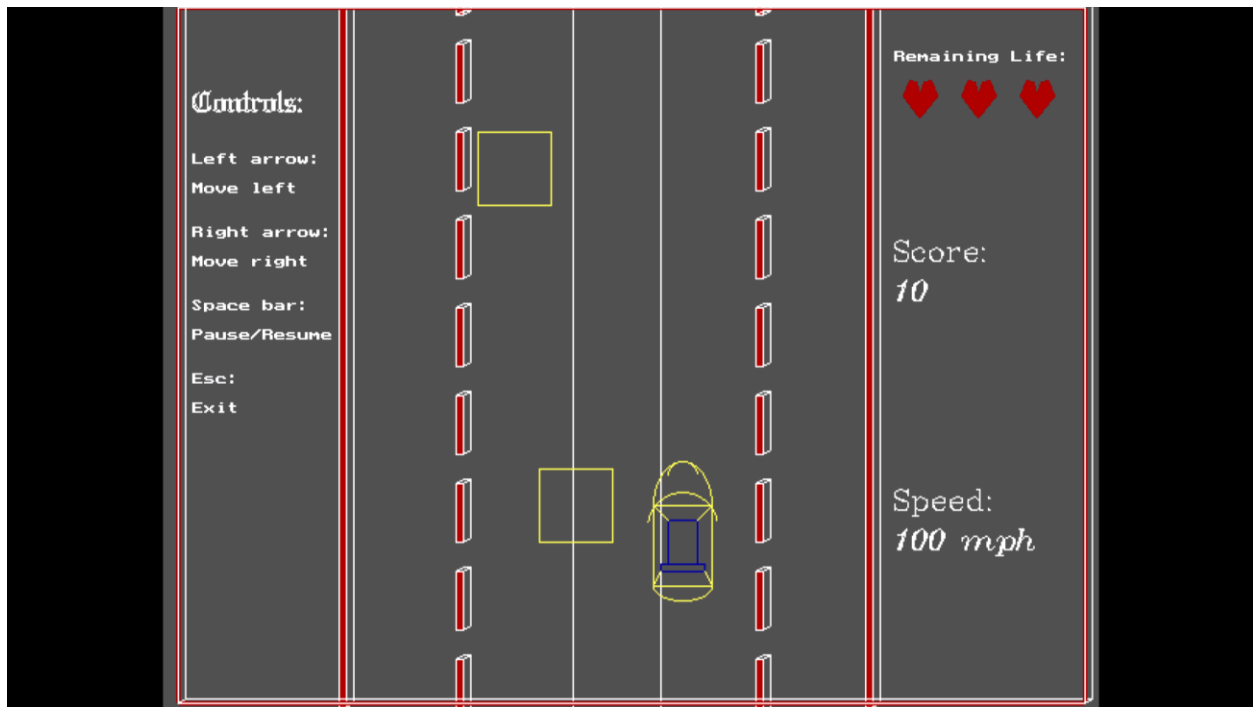


MENU

Play Game
High Score
Instructions
About
Exit



Select your car
Use arrow keys to navigate.



Current Score :60

One Chance Lost! ☹️

Press Enter to Continue!

GAME OVER

Your Score: 210

Press Enter to return to the main Menu.

High Scores

All time High Score :
780

Current High Scores :

1. 210 | 2. 0 | 3. 0 | 4. 0 | 5. 0

Press Esc to return to the Main Menu

Instructions

Objective:

Save your car from the obstacles.

Controls:

Left arrow key : Move left

Right arrow key: Move right

Up arrow key : Move Up

Down arrow key : Move Down

space bar : Pause/Resume game

Enter : Select/Choose/Continue

Esc : return to menu

Press esc to return to the main menu

About

The Game:

Hope, You have played this game and enjoyed it.
If not, then try it and score as much as you can.

Drag'N'drive, as the name suggests, you will have to drag your car away from the obstacles in order to drive further. You have to select one of the four given cars and drive it to score points by avoiding the obstacles.

Each obstacle avoided will award you 10 points. You also have 3 lives, which means you have 3 chances.

You can also give us your valuable suggestions via the details provided in the next page.

Version 1.14

About

The Developers:

This game is a part of Computer Graphics Project work. We have tried to reduce as much technical or logical errors as we can. However, it may contain some bugs.

It is developed by Rupali Singh and Aniket Kumar, students of BCA (DSVV, Haridwar).

You can mail us your valuable suggestions at:

Aniketkumar@hotmail.com

rupalisingh2709@gmail.com

The repository of this game is at: bit.ly/dragndrive

So, what are you waiting for?
Go & score as much as you can.

Press esc to return to the main menu

Task – 17

Draw a line using basic mathematical calculations ($y=mx+c$)

```
// A program in C to print a line using putpixel() function.
#include<graphics.h>
#include<stdio.h>
#include<conio.h>

int main()
{
    int gd=DETECT, gm;
    float x,y,x1,y1,x2,y2,m,c,i;

    //getting the coordinates
    printf("X1:");
    scanf("%f",&x1);
    printf("Y1:");
    scanf("%f",&y1);
    printf("X2:");
    scanf("%f",&x2);
    printf("Y2:");
    scanf("%f",&y2);

    m=(y2-y1)/(x2-x1); //slope
    c=y1-(m*x1);

    initgraph(&gd,&gm,"c:\\turbo3\\bgi");

    //line(x1,y1,x2,y2);

    if(x2<x1) //for negative values
```

```

{
    x=x1;
    x1=x2;
    x2=x;
    y=y1;
    y1=y2;
    y2=y;

    for(i=x1;i<=x2;i++)
    {
        putpixel(i,(i*m)+c,RED);
    }
}

else
{
    for(i=x1;i<=x2;i++)
    {
        putpixel(i,(i*m)+c,RED);
    }
}

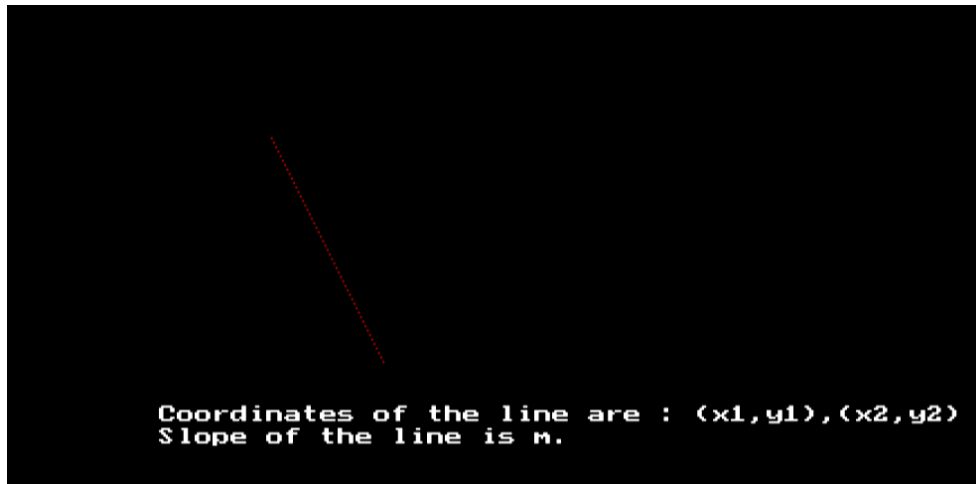
    outtextxy(0,420,"Coordinates of the line are :
(x1,y1) ,(x2,y2)");

    outtextxy(0,430,"Slope of the line is m.");

    getch();
    closegraph();
}

```


Screenshot :



Task – 18

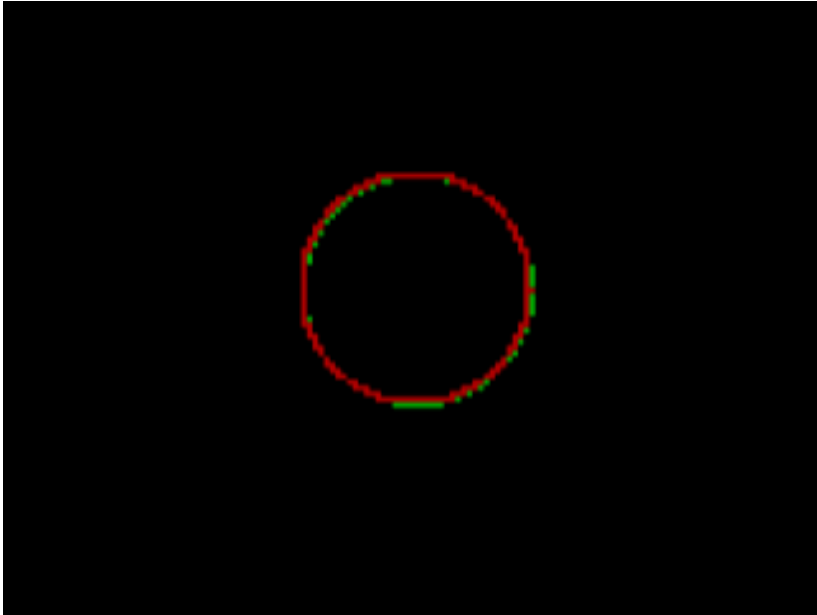
Draw a circle using trigonometric functions

```
/* A program in C to draw circle using the Putpixel() function. */
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
#include<math.h>

#define PI 3.1415

main()
{
    int gd=DETECT,gm;
    int x=50,y=50,r=20;
    float d,i;
    d=PI/180;
    initgraph(&gd,&gm,"c:\\turbo3\\bgi");
    setcolor(GREEN);
    circle(50,50,20);
    for(i=0;i<360;i++)
    {
        putpixel(x+r*cos(i*d),y+r*sin(i*d),RED);
    }
    getch();
    closegraph();
}
```

Screenshot :



Task – 19

Draw a circle and move it according to the press of arrow keys.

```
/* A program in C to print a ball and move it according to
instructions given by arrow keys. */
```

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

```
#include<process.h>
```

```
#include<dos.h>
```

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

```
#include<conio.h>
```

```
void main()
```

```
{
```

```
    int gd=DETECT,gm,i=250,j=250,ch,x=0,y=-1;
```

```
    initgraph(&gd,&gm,"c:\\turboc3\\bgi");
```

```
    while(1)
```

```
    {
```

```
        circle(i,j,10);
```

```
        outtext("Use arrow keys to control movement");
```

```
        setcolor(RED);
```

```
        if(kbhit()) //detects the pressing of key
```

```
        {
```

```
            ch=getch();
```

```
            if(ch==72) // Move up
```

```
            {
```

```
                x=0;
```

```
                y=-1;
```

```
            }
```

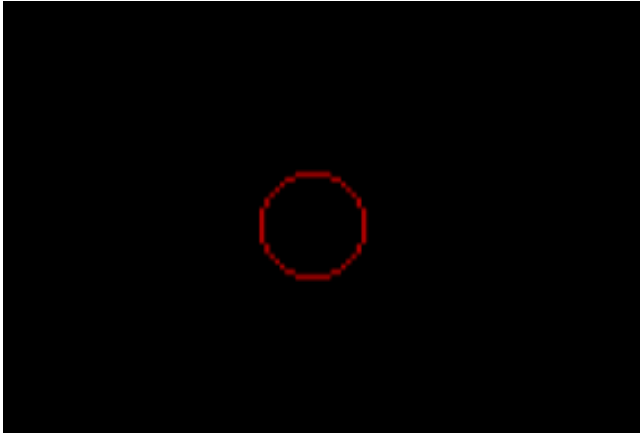
```
            if(ch==75) // Move left
```

```

    {
        x=-1;
        y=0;
    }
    if(ch==77) // Move Right
    {
        x=1;
        y=0;
    }
    if(ch==80) // Move down
    {
        x=0;
        y=1;
    }
    if(ch==27) // Esc, Exit from the screen
    {
        exit(0);
    }
    i=i+x;
    j=j+y;
    // printf("%d,%d",i,j);
    delay(5);
    cleardevice();
}
getch();
closegraph();
}

```

Screenshot :



Task 20

Write a program in C to demonstrate 2D Transformation : Scaling

```
// This a program to demonstrate scaling of an object
#include<stdio.h>
#include<conio.h>
#include<graphics.h>

void scale(float, float, float, float, float);

void main()
{
    int gd=DETECT,gm;
    float x1=50,y1=50,x2=150,y2=100,S;

    initgraph(&gd,&gm,"c:\\turboc3\\bgi");
    outtext("This is a Scaling.");
    outtextxy(50,30,"Press any key to continue");
    rectangle(x1,y1,x2,y2); //original shape
    getch();
    closegraph();

    printf("Enter the Scaling factor : ");
    printf("\nEnter value less than 1 for down scaling");
    printf("\nEnter value more than 1 for up scaling\n");
    scanf("%f",&S);

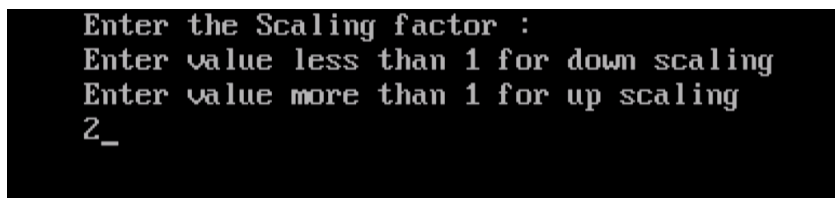
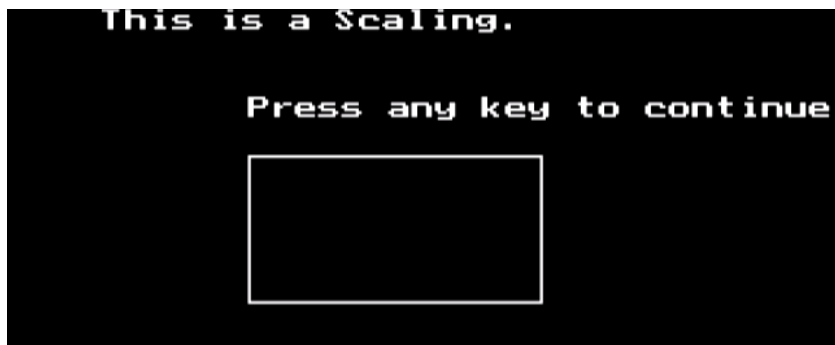
    scale(x1,y1,x2,y2,S);
    getch();
    closegraph();
}
```

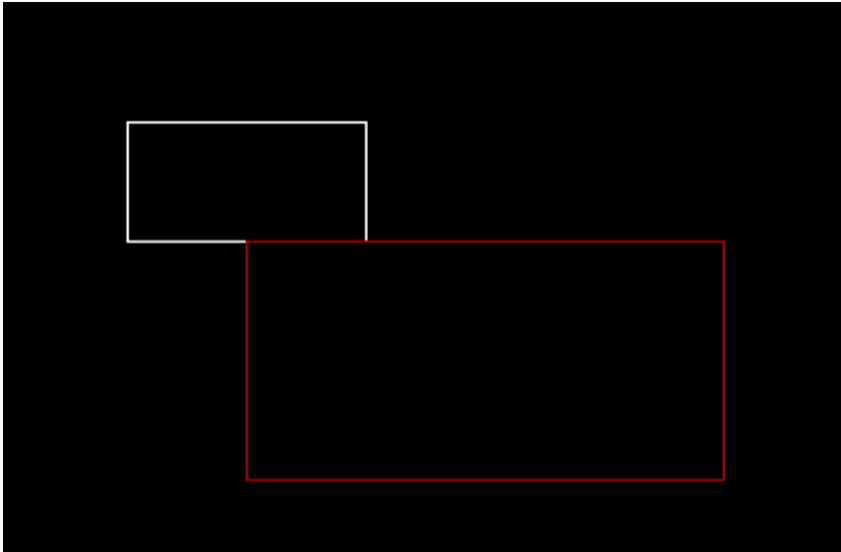
```

void scale(float x1, float y1, float x2, float y2, float S)
{
    int gd=DETECT, gm;
    initgraph(&gd, &gm, "c:\\turbo3\\bgi");
    rectangle(x1, y1, x2, y2); //original shape
    //scaling the coordinates
        x1=x1*S;
        x2=x2*S;
        y1=y1*S;
        y2=y2*S;
    setcolor(RED);          //changing the color to red
    rectangle(x1, y1, x2, y2); //translated shape
}

```

Screenshots :





```
Enter the Scaling factor :  
Enter value less than 1 for down scaling  
Enter value more than 1 for up scaling  
0.5
```

