JAGAN INSTITUTE OF MANAGEMENT STUDIES

Sector - 5, Rohini, New Delhi

(Affiliated to)

GURU GOBIND SINGH INDRAPRASTHA UNIVERSITY

SECTOR – 16 C, DWARKA, NEW DELHI

PRACTICAL FILE

COMPUTER GRAPHICS

BCA-373

Submitted to: Ms. Geeta Sharma

Professor (IT)

Submitted By: Aditya Pandey

Enrolment No.: 04814002021

BCA 3rd Year (Shift - 1)

5th Semester

Aditya Pandey (04814002021)

BCA 3rd Year 1st Shift

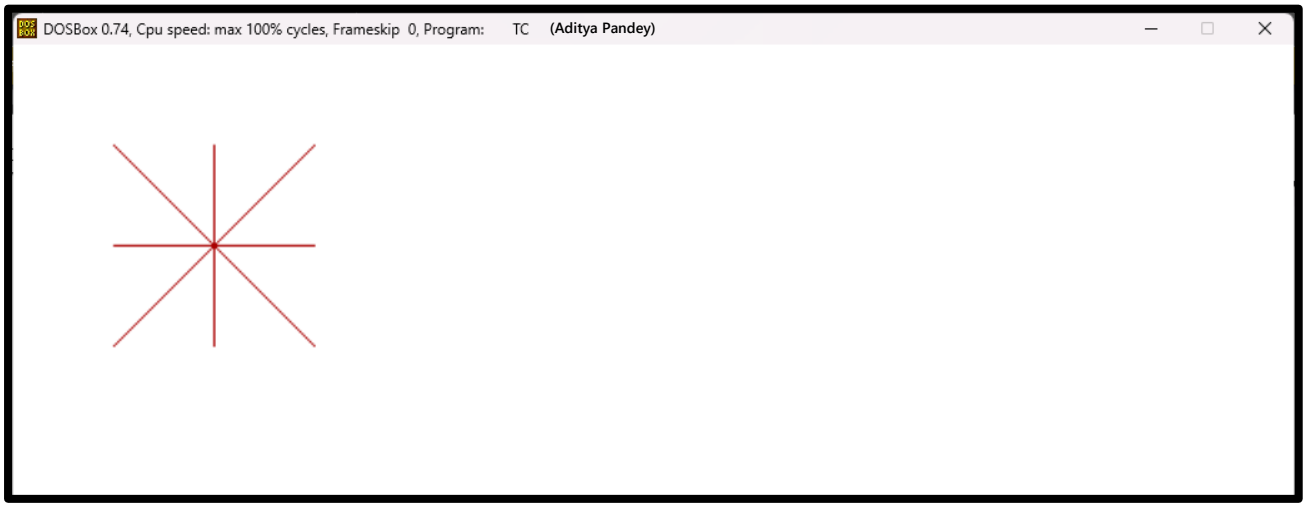
Computer Graphics (BCA 373)

Practical Assignment 1

Q1. Working with coordinates:

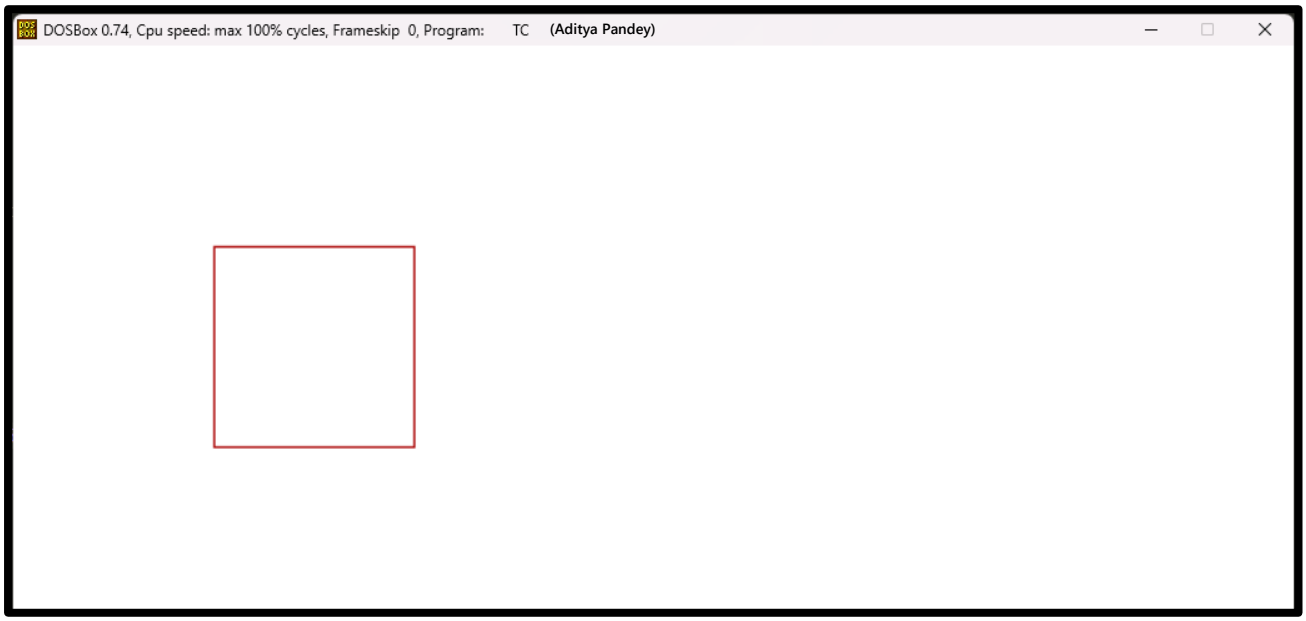
A. WAP to create * symbol.

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

void main() {
	int i;
	int gd = DETECT, gm;
	int x1 = 50, y1 = 50, x2 = 500, y2 = 350;
	initgraph(&gd, &gm, "C:\\TC\\BGI");
	line(50, 50, 150, 150);
	line(50, 150, 150, 50);
	line(100, 50, 100, 150);
	line(50, 100, 150, 100);
	getch();
	closegraph();
}
```



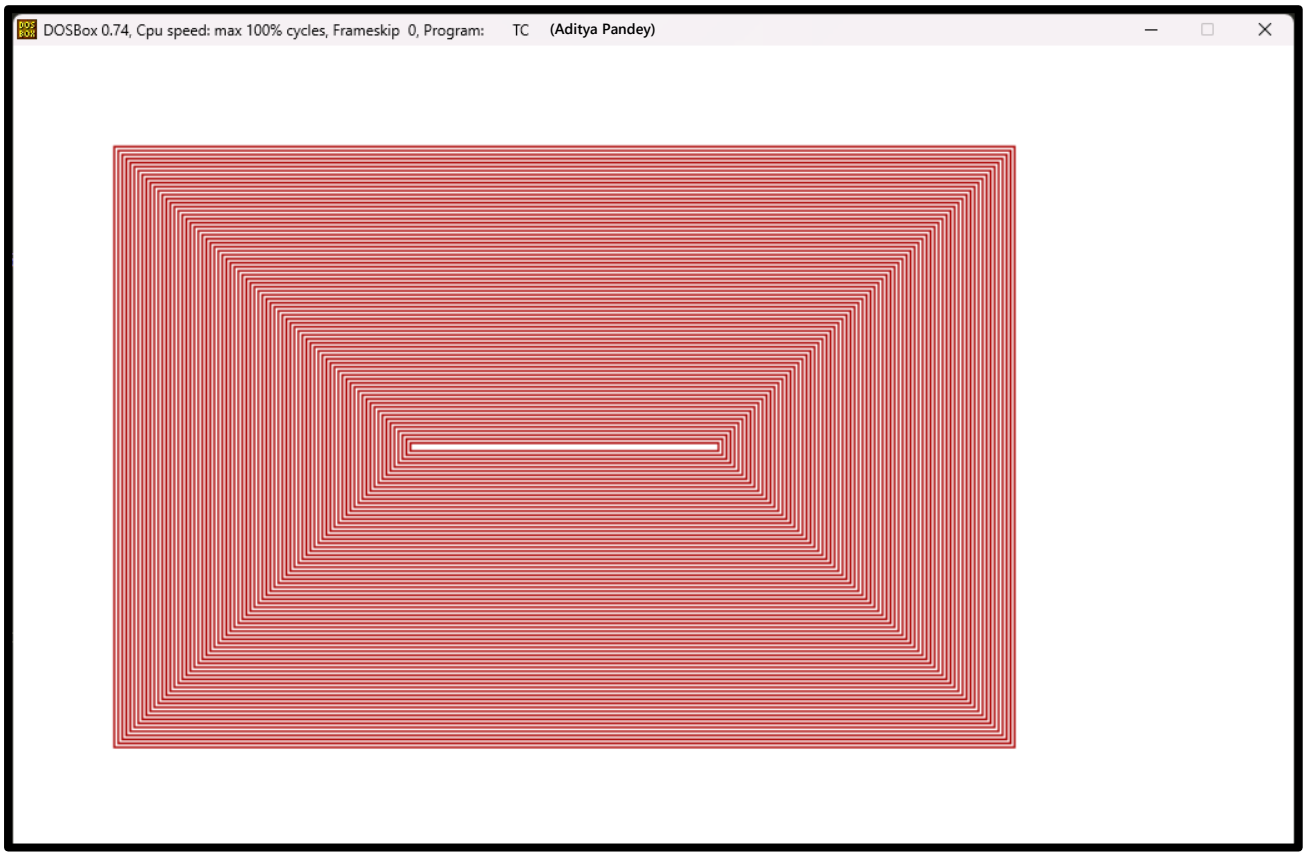
B. WAP to create a square using lines.

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

void main() {
	int i;
	int gd = DETECT, gm;
	int x1 = 50, y1 = 50, x2 = 500, y2 = 350;
	initgraph(&gd, &gm, "C:\\TC\\BGI");
	line(100, 100, 200, 100);
	line(200, 100, 200, 200);
	line(200, 200, 100, 200);
	line(100, 200, 100, 100);
	getch();
	closegraph();
}
```



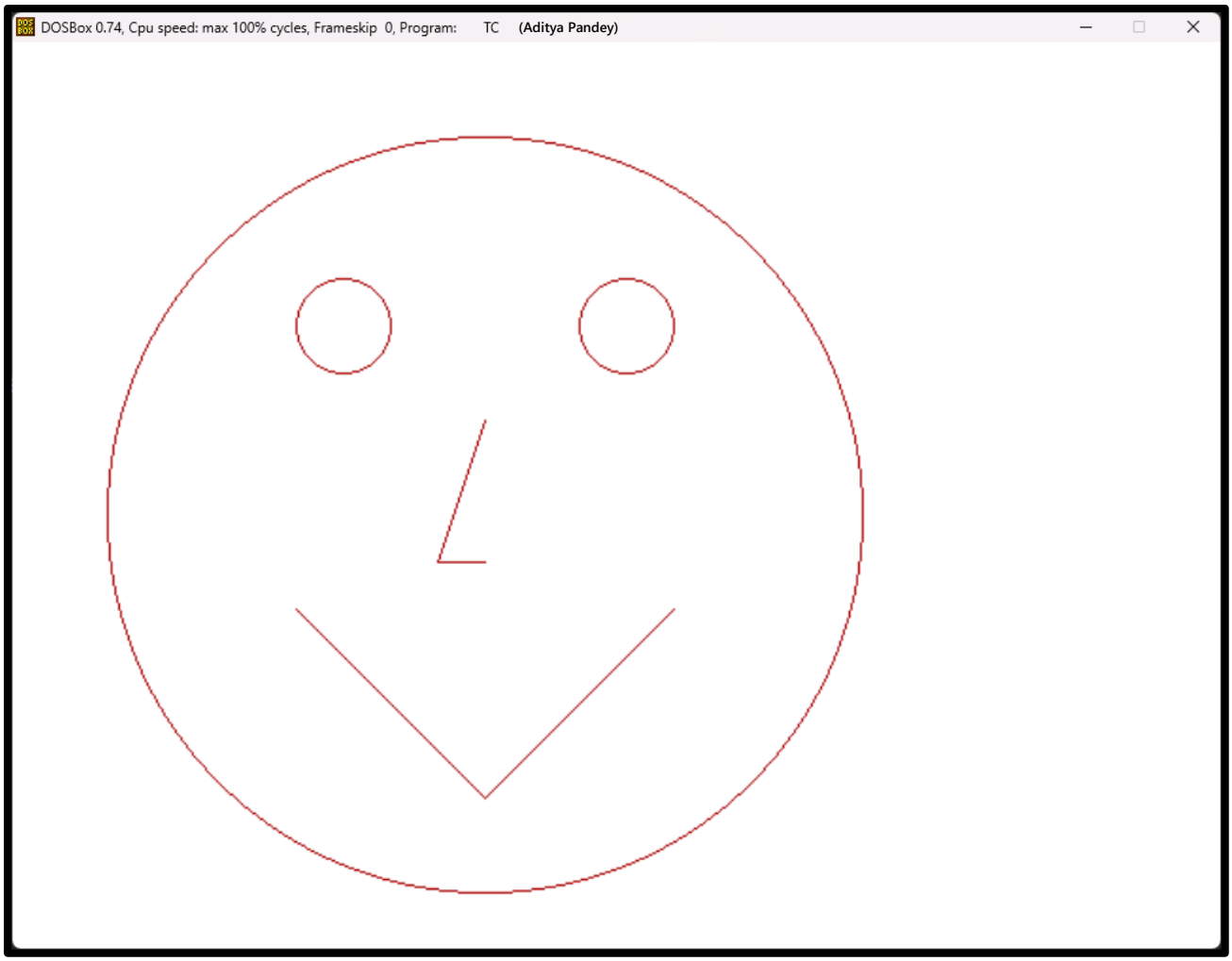
C. WAP to create rectangle inside rectangle.

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

void main() {
	int i;
	int gd = DETECT, gm;
	int x1 = 50, y1 = 50, x2 = 500, y2 = 350;
	initgraph(&gd, &gm, "C:\\TC\\BGI");
	for(i=0 ; i<150 ; i+=2) {
		rectangle(x1, y1, x2, y2);
		x1+=2; y1+=2; x2-=2; y2-=2;
	}
	getch();
	closegraph();
}
```



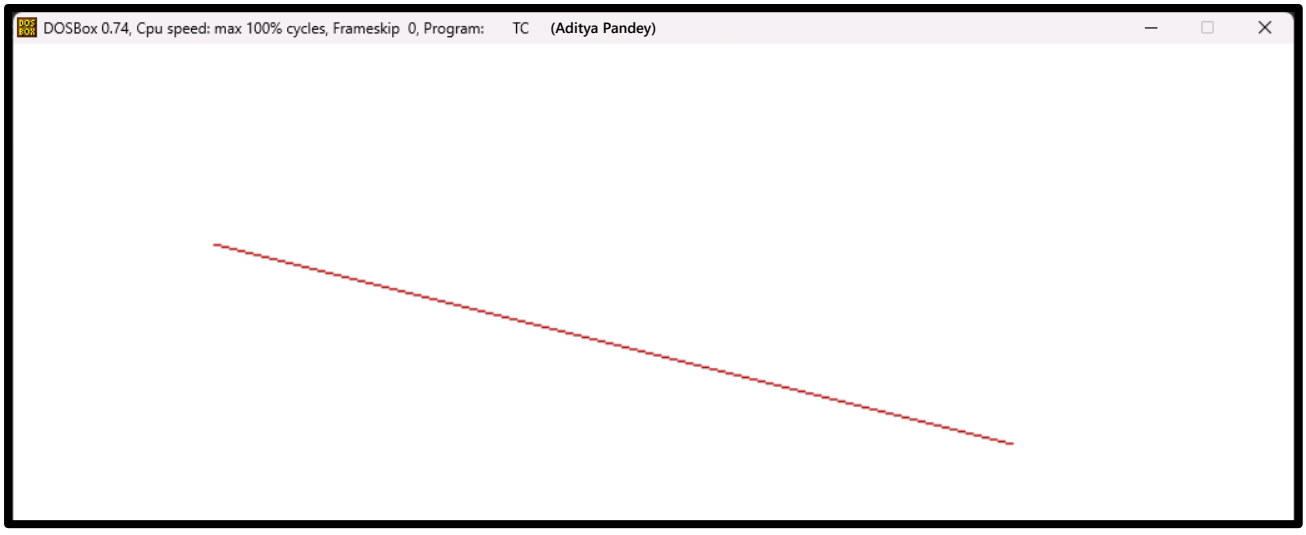
D. WAP to create a smiley.

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

void main() {
	int i;
	int gd = DETECT, gm;
	int x1 = 50, y1 = 50, x2 = 500, y2 = 350;
	initgraph(&gd, &gm, "C:\\TC\\BGI");
	circle(250, 250, 200);
	circle(175, 150, 25);
	circle(325, 150, 25);
	line(250, 200, 225, 275);
	line(225, 275, 250, 275);
	line(150, 300, 250, 400);
	line(250, 400, 350, 300);
	getch();
	closegraph();
}
```



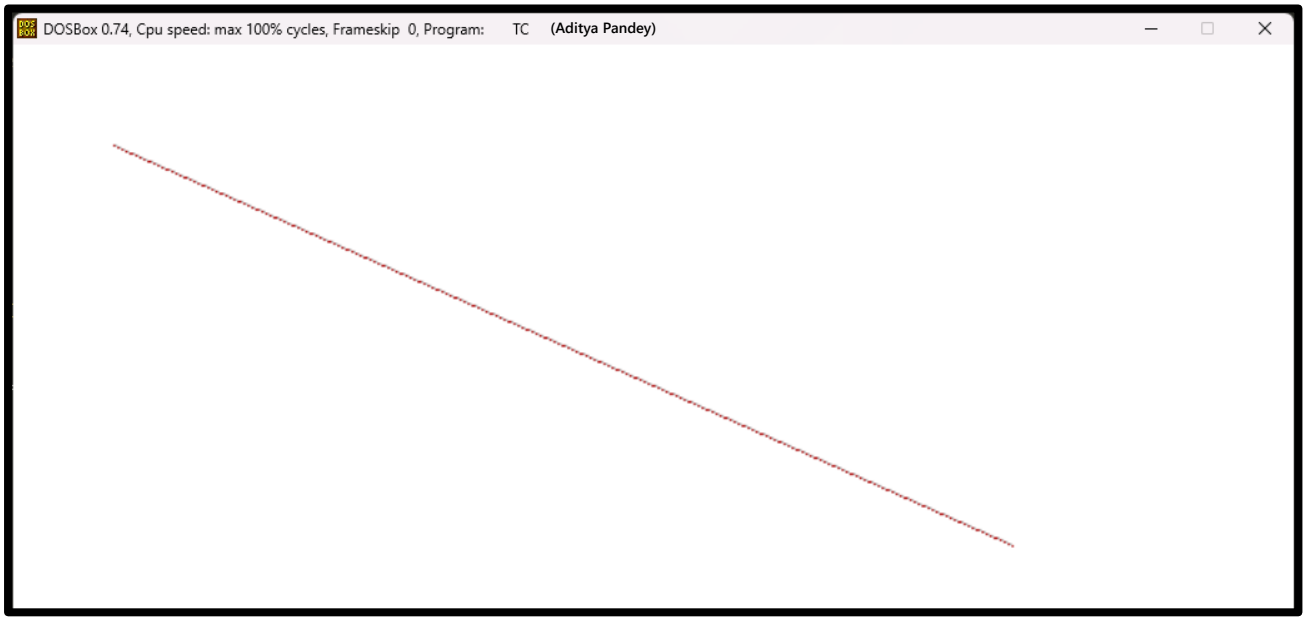
Q2. WAP to scan convert a line using DDA Line Drawing Alogrithm.

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

void main() {
	int gd = DETECT, gm;
	float x, y, dx, dy, steps;
	int x0, x1, y0, y1, i;
	initgraph(&gd, &gm, "C:\\TC\\BGI");
	setbkcolor(WHITE);
	x0 = 100, y0 = 100, x1 = 500, y1 = 200;
	dx = (float)(x1-x0);
	dy = (float)(y1-y0);
	if(dx>=dy) {
		steps = dx;
	} else {
		steps = dy;
	}
	dx = dx/steps;
	dy = dy/steps;
	x = x0;
	y = y0;
	i = 1;
	while(i<=steps) {
		putpixel(x, y, RED);
		x += dx;
		y += dy;
		i += 1;
	}
	getch();
	closegraph();
}
```



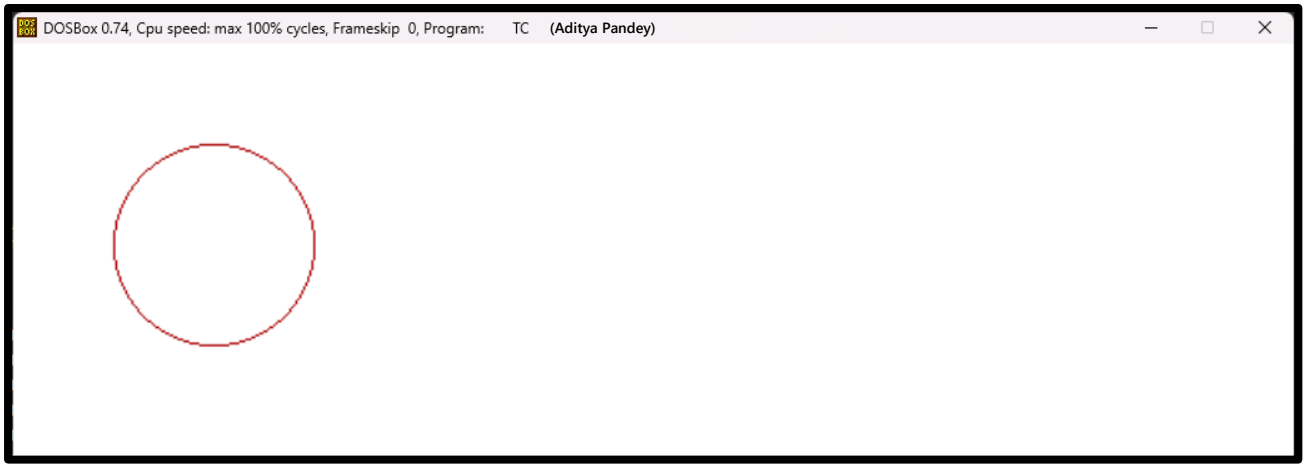
Q3. WAP to scan convert a line using Bresenham's Line Drawing Alogrithm.

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

void main() {
	int gd = DETECT, gm;
	float x, y, dx, dy, steps, d;
	int x0, x1, y0, y1;
	initgraph(&gd, &gm, "C:\\TC\\BGI");
	setbkcolor(WHITE);
	x0 = 50, y0 = 50, x1 = 500, y1 = 250;
	dx = (float)(x1-x0);
	dy = (float)(y1-y0);
	x = x0;
	y = y0;
	d = 2*dy - dx;
	while(x < x1) {
		if(d >= 0) {
			putpixel(x, y, 7);
			y = y + 1;
			d  = d + 2*dy - 2*dx;
		} else {
			putpixel(x, y, RED);
			d = d + 2*dy;
		}
		x += 1;
	}
	getch();
	closegraph();
}
```



Q4. WAP to scan convert a circle using Bresenham's Circle Drawing Algorithm.

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

void main() {
	int gd = DETECT, gm;
	int x0=100, y0=100, r=50;
	int x = 0;
	int y = r;
	int d = 3 - 2 * r;
	initgraph(&gd, &gm, "C:\\TC\\BGI");
	setbkcolor(WHITE);
	while (x <= y) {
		putpixel(x0 + x, y0 + y, RED);
		putpixel(x0 - x, y0 + y, RED);
		putpixel(x0 + x, y0 - y, RED);
		putpixel(x0 - x, y0 - y, RED);
		putpixel(x0 + y, y0 + x, RED);
		putpixel(x0 - y, y0 + x, RED);
		putpixel(x0 + y, y0 - x, RED);
		putpixel(x0 - y, y0 - x, RED);
		if (d < 0) {
			d += 4*x + 6;
		} else {
			d += 4*(x-y) + 10;
			y -= 1;
		}
		x++;
	}
	getch();
	closegraph();
}
```



Aditya Pandey (04814002021)

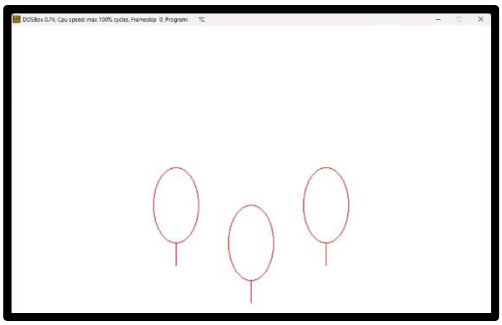
BCA 3rd Year 1st Shift

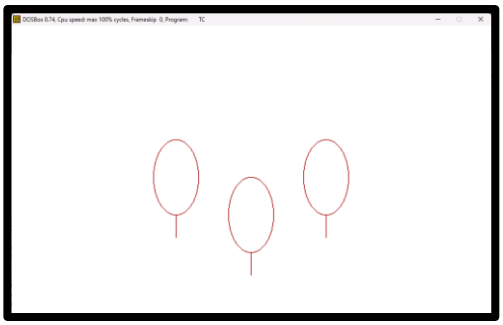
Computer Graphics (BCA 373)

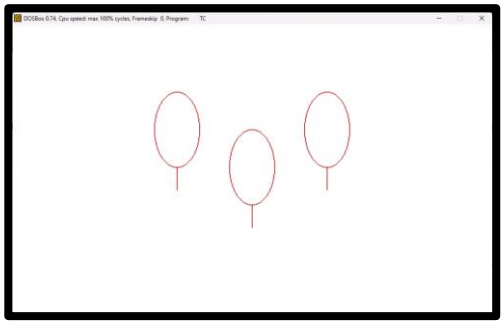
Practical Assignment 2

Q1. WAP to draw Flying Balloon.

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

void main() {
	int i, j;
	int gd = DETECT, gm;
	initgraph(&gd, &gm, "C:\\TC\\BGI");
	setbkcolor(15);
	j=getmaxy()/2;
	for(i=0; i<100; i+=1) {
		cleardevice();
		setcolor(RED);
		ellipse(getmaxx()/2-100, j, 0, 360, 30, 50);
		line(getmaxx()/2-100, j+50, getmaxx()/2-100, j+80);
		ellipse(getmaxx()/2, j+50, 0, 360, 30, 50);
		line(getmaxx()/2, j+100, getmaxx()/2, j+130);
		ellipse(getmaxx()/2+100, j, 0, 360, 30, 50);
		line(getmaxx()/2+100, j+50, getmaxx()/2+100, j+80);
		j-=1;
		delay(10);
	}
	getch();
	closegraph();
}
```

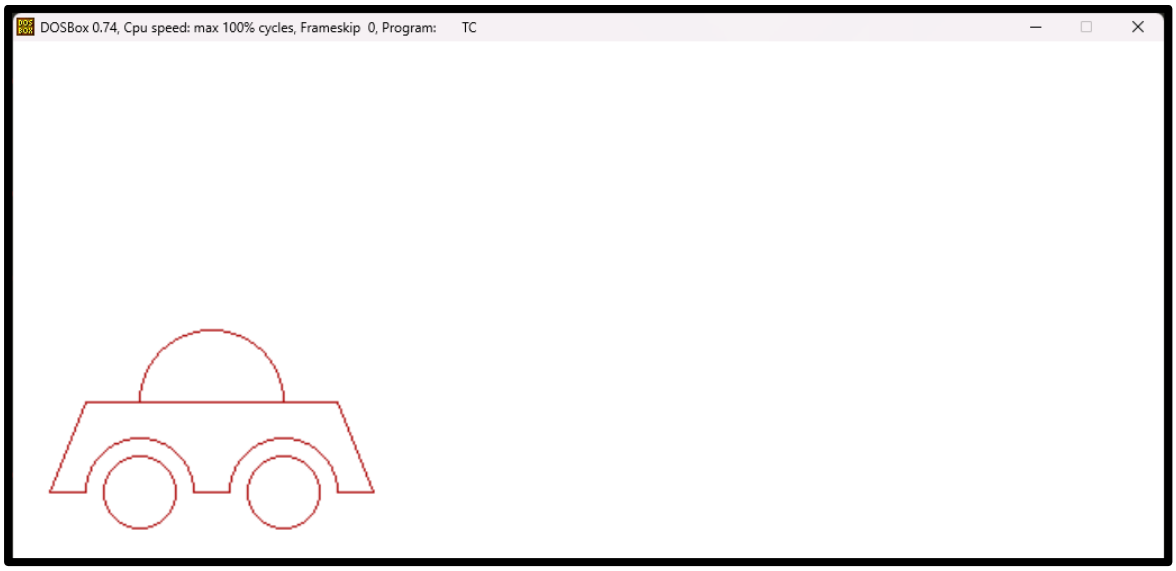


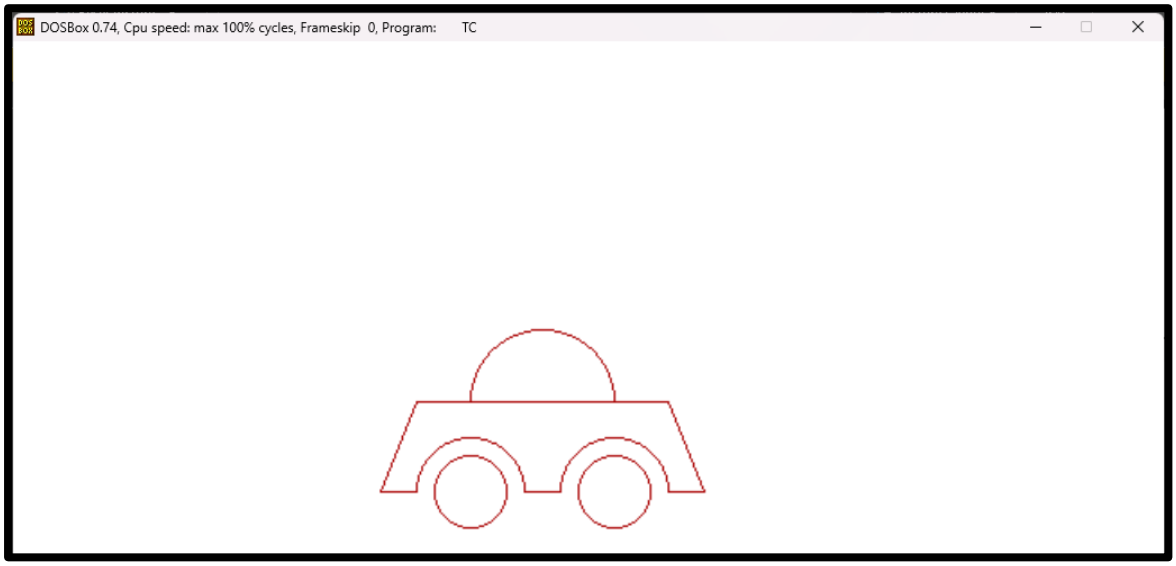


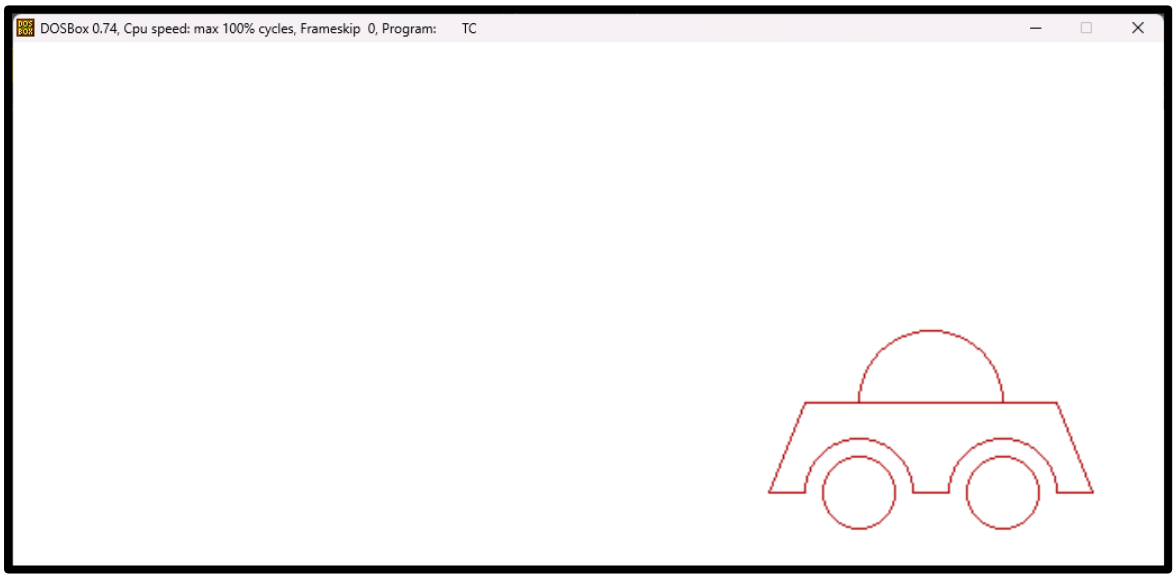


Q2. WAP to move a vehicle (car) from left to right.

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

void main() {
	int i, j;
	int gd = DETECT, gm;
	initgraph(&gd, &gm, "C:\\TC\\BGI");
	setbkcolor(15);
	j=20;
	for(i=0; i<100; i+=1) {
		cleardevice();
		setcolor(RED);
		line(j, 250, j+20, 250);
		ellipse(j+50, 250, 0, 180, 30, 30);
		circle(j+50, 250, 20);
		line(j+80, 250, j+100, 250);
		ellipse(j+130, 250, 0, 180, 30, 30);
		circle(j+130, 250, 20);
		line(j+160, 250, j+180, 250);
		line(j, 250, j+20, 200);
		line(j+20, 200, j+160, 200);
		line(j+160, 200, j+180, 250);
		ellipse(j+90, 200, 0, 180, 40, 40);
		j+=4;
		delay(10);
	}
	getch();
	closegraph();
}
```



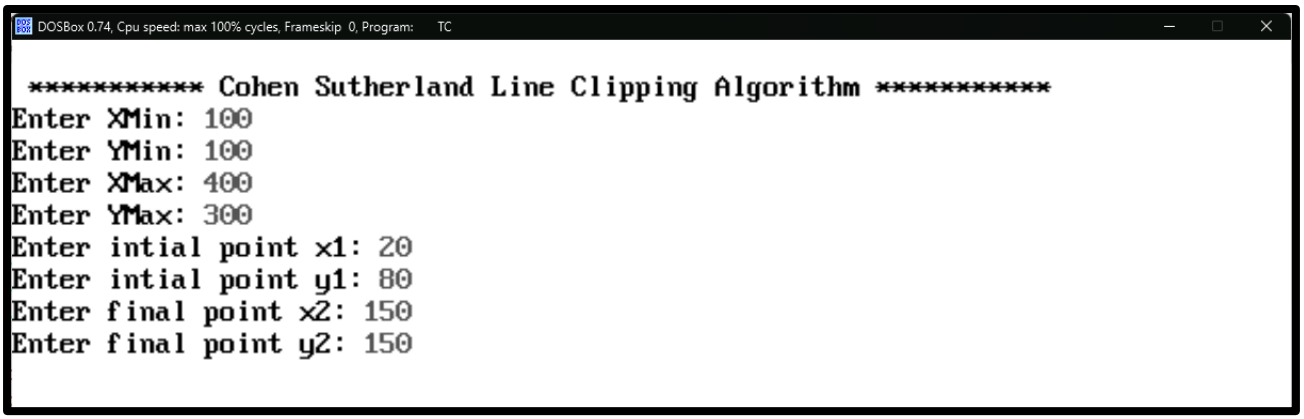

DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC

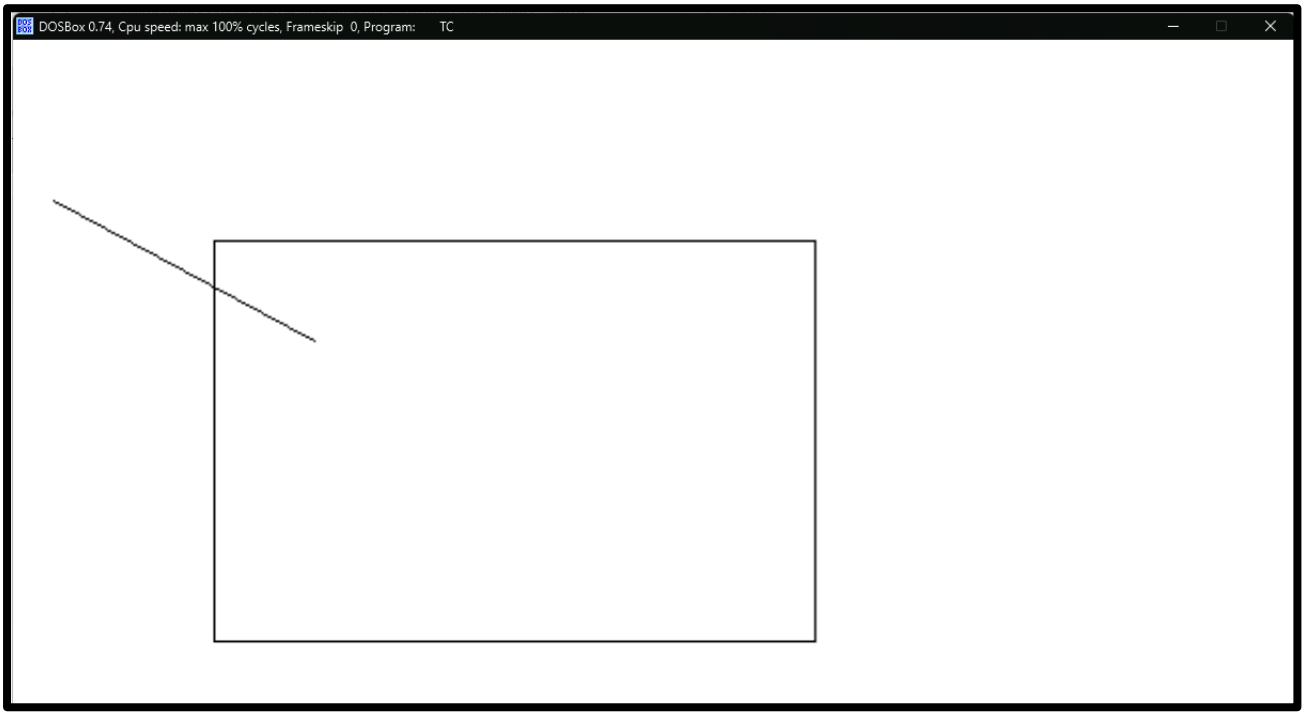

DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC

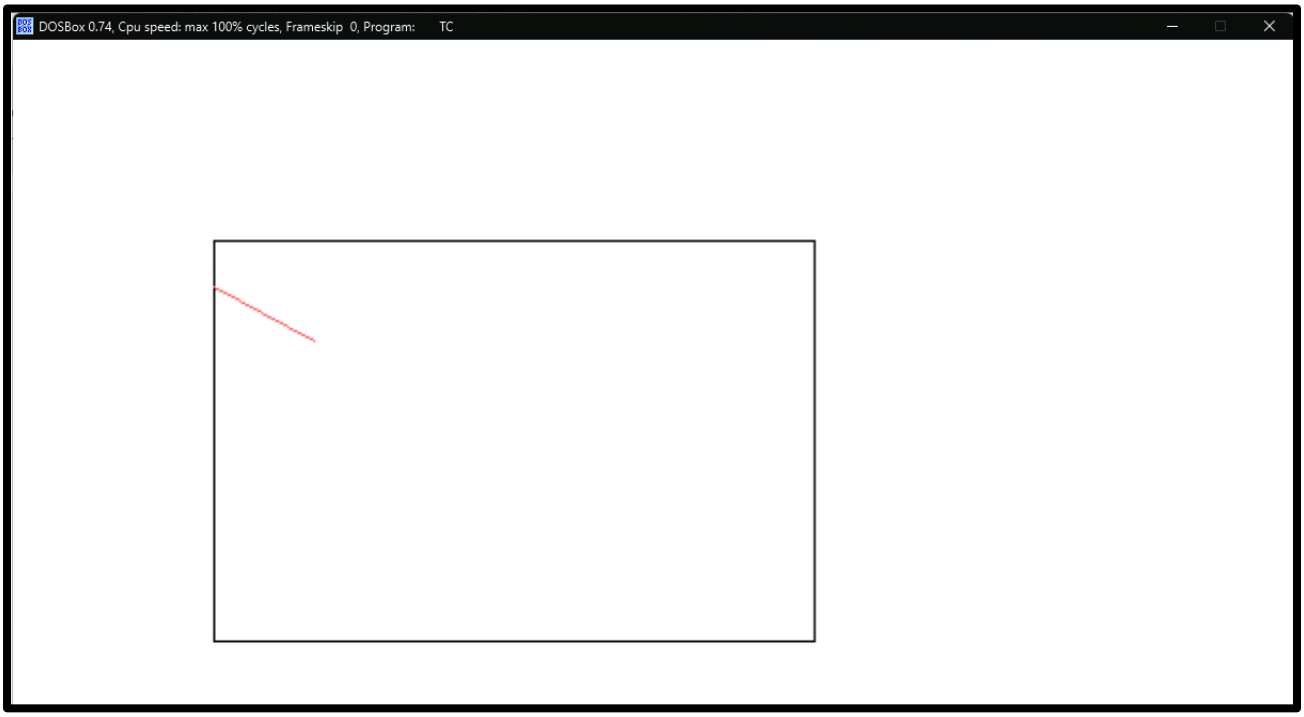
Q3. WAP to clip a line segment using Cohen-Sutherland algorithm.

```
#include<graphics.h>
#include<conio.h>
#include<stdio.h>
int codeEntry[4] = {0,0,0,0}, codeExit[4] = {0,0,0,0}, codeOper[4] = {0,0,0,0},
codeOperFlag=0, i;
float slope;
int XMin, YMin, XMax, YMax, x1, y1, x2, y2;
void codeLine(int calcCode[4], float x, float y, float XMin, float YMin, float XMax,
float YMax) {
	if(x < XMin) {
		calcCode[3] = 1;		// Left
	} if(x > XMax) {
		calcCode[2] = 1;		// Right
	} if(y > YMax) {
		calcCode[1] = 1;		// Bottom
	} if(y < YMin) {
		calcCode[0] = 1;		// Top
	}
}

void main() {
	int gd=DETECT, gm;
	initgraph(&gd, &gm, "C:\\TC\\BGI");
	setbkcolor(WHITE);
	printf("\n *********** Cohen Sutherland Line Clipping Algorithm ********** \n");
	printf("Enter XMin: ");
	scanf("%d", &XMin);
	printf("Enter YMin: ");
	scanf("%d", &YMin);
	printf("Enter XMax: ");
	scanf("%d", &XMax);
	printf("Enter YMax: ");
	scanf("%d", &YMax);
	printf("Enter intial point x1: ");
	scanf("%d",&x1);
	printf("Enter intial point y1: ");
	scanf("%d",&y1);
	printf("Enter final point x2: ");
	scanf("%d",&x2);
	printf("Enter final point y2: ");
	scanf("%d",&y2);
	delay(1000);
	cleardevice();
	setcolor(BLACK);
	rectangle(XMin, YMin, XMax, YMax);
	line(x1, y1, x2, y2);
	codeLine(codeEntry, x1, y1, XMin, YMin, XMax, YMax);
	codeLine(codeExit, x2, y2, XMin, YMin, XMax, YMax);
	codeOperFlag = 1;
	for(i=0 ; i<4 ; i++) {
		codeOper[i] = codeEntry[i] || codeExit[i];
		if(codeOper[i] == 1) { codeOperFlag = 0; }
	}

	if(codeOperFlag == 1) { printf("Case fully visible."); }
	else {
		codeOperFlag = 1;
		for(i=0 ; i<4 ; i++) {
			codeOper[i] = codeEntry[i] && codeExit[i];
```

```
                        if(codeOper[i] == 1) { codeOperFlag = 0; }
                }
                if(codeOperFlag == 0) { printf("Case fully invisible."); }
                else {
                        slope = (float)(y2-y1)/(x2-x1);
                        if(codeEntry[3] == 1 && (x1<XMin || x1>XMax)) {
                                y1 += (XMin-x1)*slope;
                                x1 = XMin;
                        } if(codeEntry[2] == 1 && (x1<XMin || x1>XMax)) {
                                y1 += (XMax-x1)*slope;
                                x1 = XMax;
                        } if(codeEntry[1] == 1 && (y1<YMin || y1>YMax)) {
                                x1 += (YMax-y1)/slope;
                                y1 = YMax;
                        } if(codeEntry[0] == 1 && (y1<YMin || y1>YMax)) {
                                x1 += (YMin-y1)/slope;
                                y1 = YMin;
                        }
                        if(codeExit[3] == 1 && (x2<XMin || x2>XMax)) {
                                y2 += (XMin-x2)*slope;
                                x2 = XMin;
                        } if(codeExit[2] == 1 && (x2<XMin || x2>XMax)) {
                                y2 += (XMax-x2)*slope;
                                x2 = XMax;
                        } if(codeExit[1] == 1 && (y2<YMin || y2>YMax)) {
                                x2 += (YMax-y2)/slope;
                                y2 = YMax;
                        } if(codeExit[0] == 1 && (y2<YMin || y2>YMax)) {
                                x2 += (YMin-y2)/slope;
                                y2 = YMin;
                        }
                        delay(3000);
                        clearviewport();
                        rectangle(XMin, YMin, XMax, YMax);
                        setcolor(RED);
                        line(x1, y1, x2, y2);
                }
        }
        getch();
        closegraph();
}
```




DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC


DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC

Aditya Pandey (04814002021)

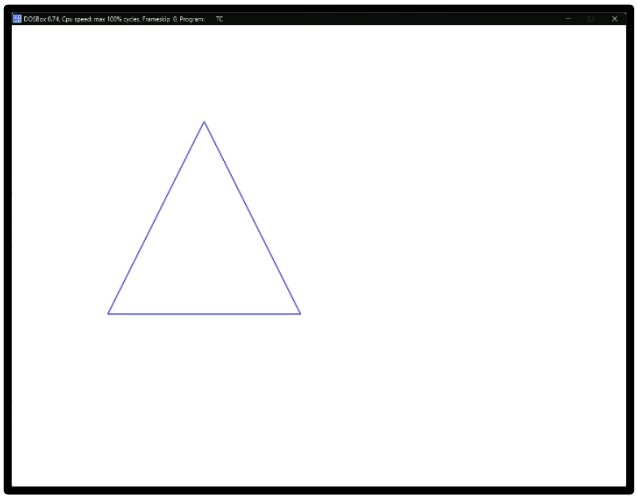
BCA 3rd Year 1st Shift

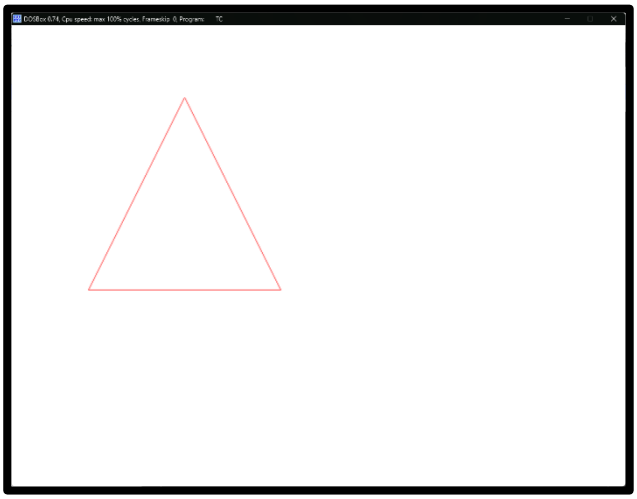
Computer Graphics (BCA 373)

Practical Assignment 3

Q1. WAP to translate a triangle in 2D plane.

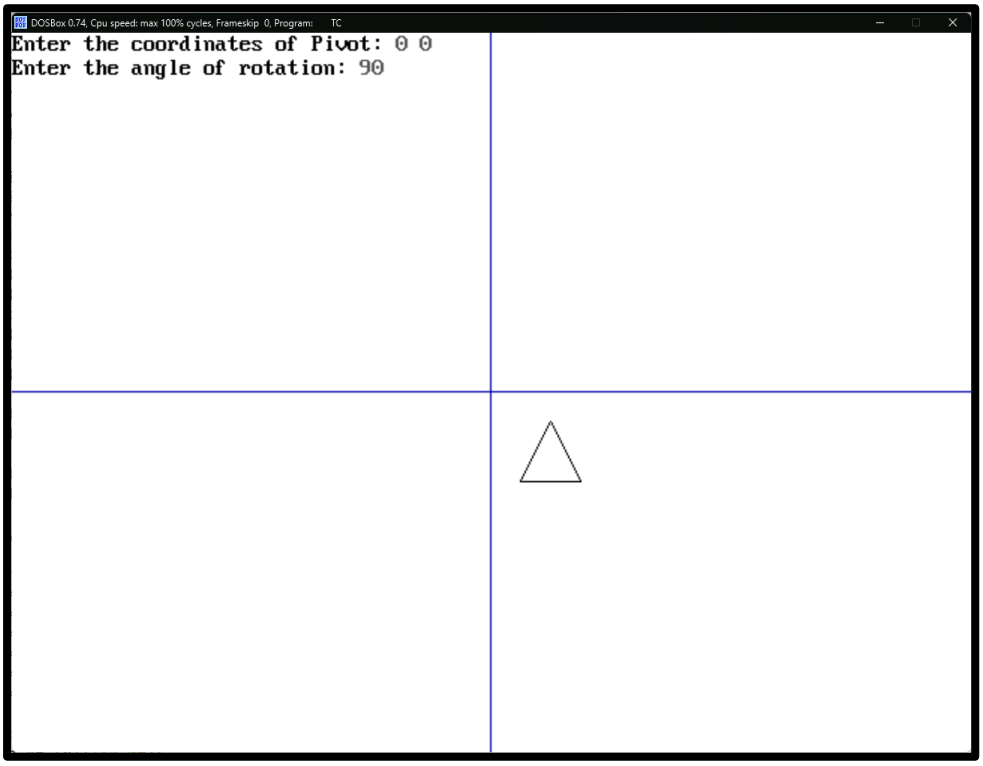
```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>

void main() {
	int x1=200, y1=100, x2=100, y2=300, x3=300, y3=300, dx=-20, dy=-25;
	int gd = DETECT, gm;
	initgraph(&gd, &gm, "C:\\TC\\BGI");
	setbkcolor(WHITE);
	setcolor(BLUE);
	line(x1, y1, x2, y2);
	line(x2, y2, x3, y3);
	line(x3, y3, x1, y1);
	delay(1000);
	x1 += dx;
	y1 += dy;
	x2 += dx;
	y2 += dy;
	x3 += dx;
	y3 += dy;
	cleardevice();
	setcolor(RED);
	line(x1, y1, x2, y2);
	line(x2, y2, x3, y3);
	line(x3, y3, x1, y1);
	getch();
	closegraph();
}
```

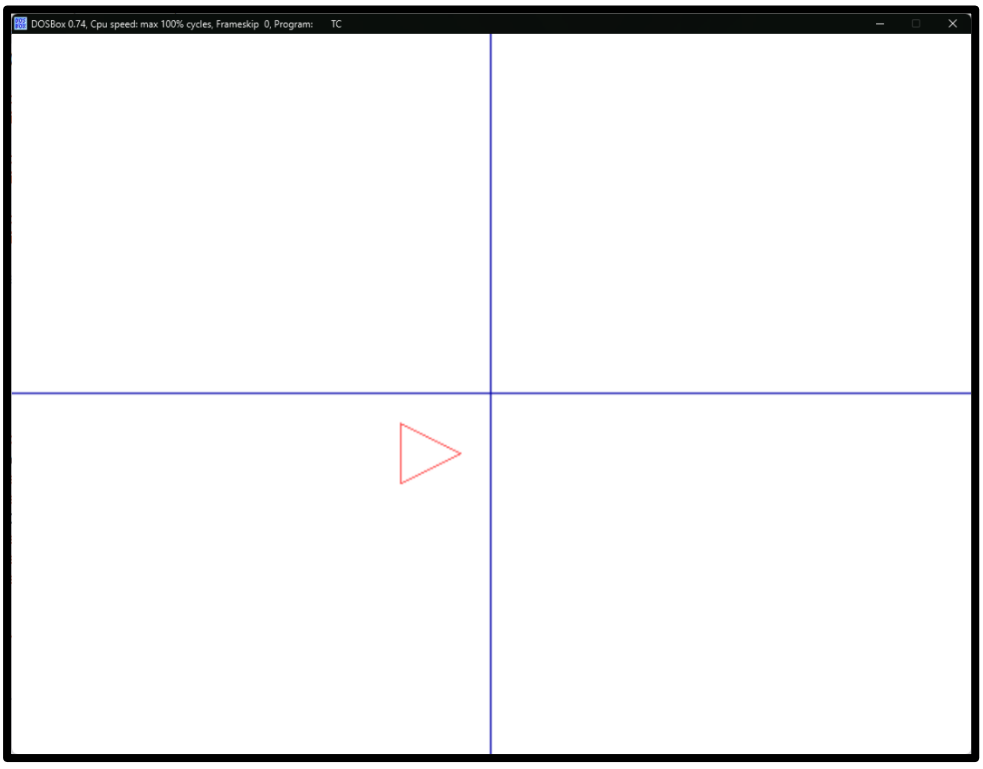




Q2. WAP to rotate a triangle in 2D plane.

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
#include<math.h>
#define pi 3.14f

void main() {
	int x1=40, y1=20, x2=20, y2=60, x3=60, y3=60, xc=0, yc=0;
	float angle=60.0f, a1, b1, a2, b2, a3, b3;

	int gd = DETECT, gm;
	initgraph(&gd, &gm, "C:\\TC\\BGI");
	setbkcolor(WHITE);
	setcolor(BLUE);
	line(getmaxx() / 2, 0, getmaxx() / 2, getmaxy());
	line(0, getmaxy() / 2, getmaxx(), getmaxy() / 2);
	setcolor(BLACK);
	line(x1+getmaxx()/2, y1+getmaxy()/2, x2+getmaxx()/2, y2+getmaxy()/2);
	line(x2+getmaxx()/2, y2+getmaxy()/2, x3+getmaxx()/2, y3+getmaxy()/2);
	line(x3+getmaxx()/2, y3+getmaxy()/2, x1+getmaxx()/2, y1+getmaxy()/2);
	printf("Enter the coordinates of Pivot: ");
	scanf("%d %d", &xc, &yc);
	printf("Enter the angle of rotation: ");
	scanf("%f", &angle);

	delay(1000);
	x1 -= xc;    y1 -= yc;
	x2 -= xc;    y2 -= yc;
	x3 -= xc;    y3 -= yc;

	angle *= (pi/180.0f);
	a1 = x1*cos(angle) - y1*sin(angle);
	b1 = x1*sin(angle) + y1*cos(angle);
	a2 = x2*cos(angle) - y2*sin(angle);
	b2 = x2*sin(angle) + y2*cos(angle);
	a3 = x3*cos(angle) - y3*sin(angle);
	b3 = x3*sin(angle) + y3*cos(angle);

	a1 += xc;    b1 += yc;
	a2 += xc;    b2 += yc;
	a3 += xc;    b3 += yc;

	cleardevice();
	setcolor(BLUE);
	line(getmaxx() / 2, 0, getmaxx() / 2, getmaxy());
	line(0, getmaxy() / 2, getmaxx(), getmaxy() / 2);
	setcolor(RED);
	line(a1+getmaxx()/2, b1+getmaxy()/2, a2+getmaxx()/2, b2+getmaxy()/2);
	line(a2+getmaxx()/2, b2+getmaxy()/2, a3+getmaxx()/2, b3+getmaxy()/2);
	line(a3+getmaxx()/2, b3+getmaxy()/2, a1+getmaxx()/2, b1+getmaxy()/2);
	getch();
	closegraph();
}
```

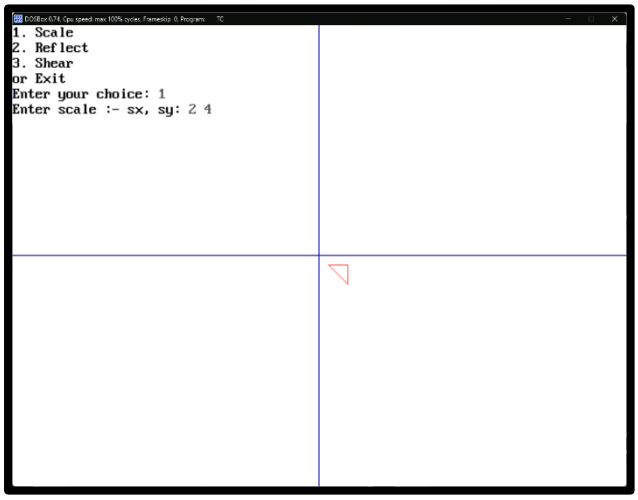

DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC
Enter the coordinates of Pivot: 0 0
Enter the angle of rotation: 90

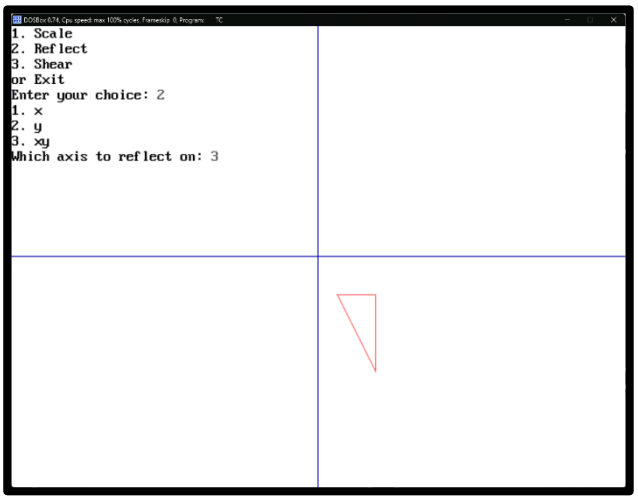

DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC

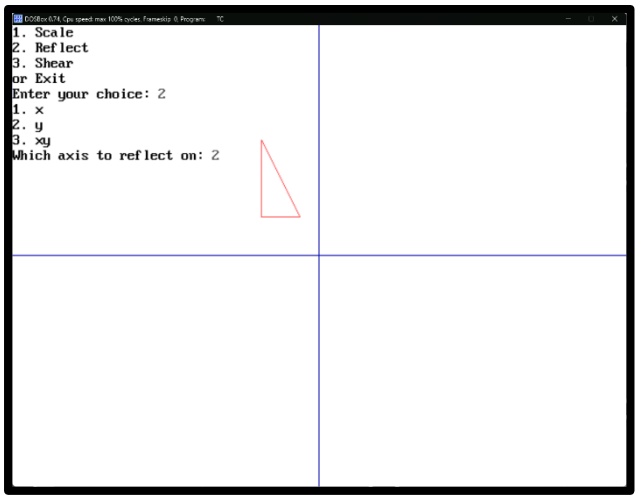
Q3. Write a menu driven program to scale, reflect, and shear a triangle in a 2D plane.

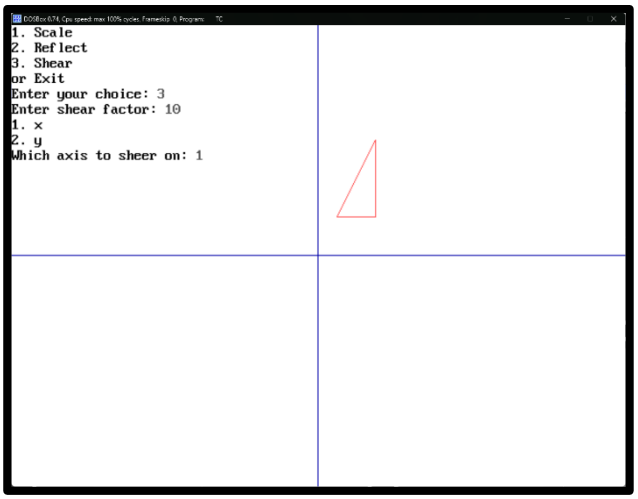
```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
void main() {
	int x1=10, y1=10, x2=30, y2=10, x3=30, y3=30, dx=20, dy=30, axis, ch=0;
	float sh=0.2;
	int gd = DETECT, gm;
	initgraph(&gd, &gm, "C:\\TC\\BGI");
	setbkcolor(WHITE);
	do {
		delay(1000);
		clrscr();
		delay(250);
		setcolor(BLUE);
		line(getmaxx() / 2, 0, getmaxx() / 2, getmaxy());
		line(0, getmaxy() / 2, getmaxx(), getmaxy() / 2);
		setcolor(RED);
		line(x1+getmaxx()/2, y1+getmaxy()/2, x2+getmaxx()/2, y2+getmaxy()/2);
		line(x2+getmaxx()/2, y2+getmaxy()/2, x3+getmaxx()/2, y3+getmaxy()/2);
		line(x3+getmaxx()/2, y3+getmaxy()/2, x1+getmaxx()/2, y1+getmaxy()/2);
		printf("1. Scale\n2. Reflect\n3. Shear\nor Exit\nEnter your choice: ");
		scanf("%d", &ch);
		if(ch==1) {
			printf("Enter scale :- sx, sy: ");
			scanf("%d %d", &dx, &dy);
			x1 *= dx;	y1 *= dy;
			x2 *= dx;	y2 *= dy;
			x3 *= dx;	y3 *= dy;
		} else if(ch==2) {
			printf("1. x\n2. y\n3. xy\nWhich axis to reflect on: ");
			scanf("%d", &axis);
			if(axis==1) {
				dx = 1;		dy = -1;
			} else if(axis==2) {
				dx = -1;	dy = 1;
			} else if(axis==3) {
				dx = -1;	dy = -1;
			} else {
				printf("Invalid choice");
				continue;
			}
			x1 *= dx;	y1 *= dy;
			x2 *= dx;	y2 *= dy;
			x3 *= dx;	y3 *= dy;
		} else if(ch==3) {
			printf("Enter shear factor: ");
			scanf("%d", &sh);
			printf("1. x\n2. y\nWhich axis to sheer on: ");
			scanf("%d", &axis);
			if(axis==1) {
				x1 += y1*sh;
				x2 += y2*sh;
				x3 += y3*sh;
			} else if(axis==2) {
				y1 += x1*sh;
				y2 += x2*sh;
				y3 += x3*sh;
			} else {
				printf("Invalid choice");
```

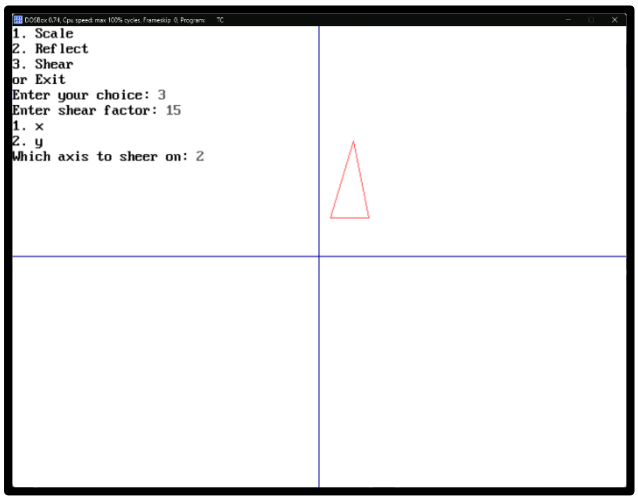
```
                        continue;
                    }
                } else {
                    break;
                }
            } while(ch);
            closegraph();
    }
```

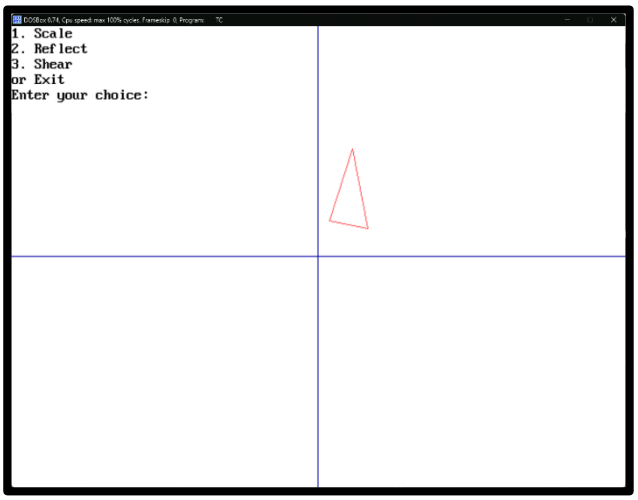








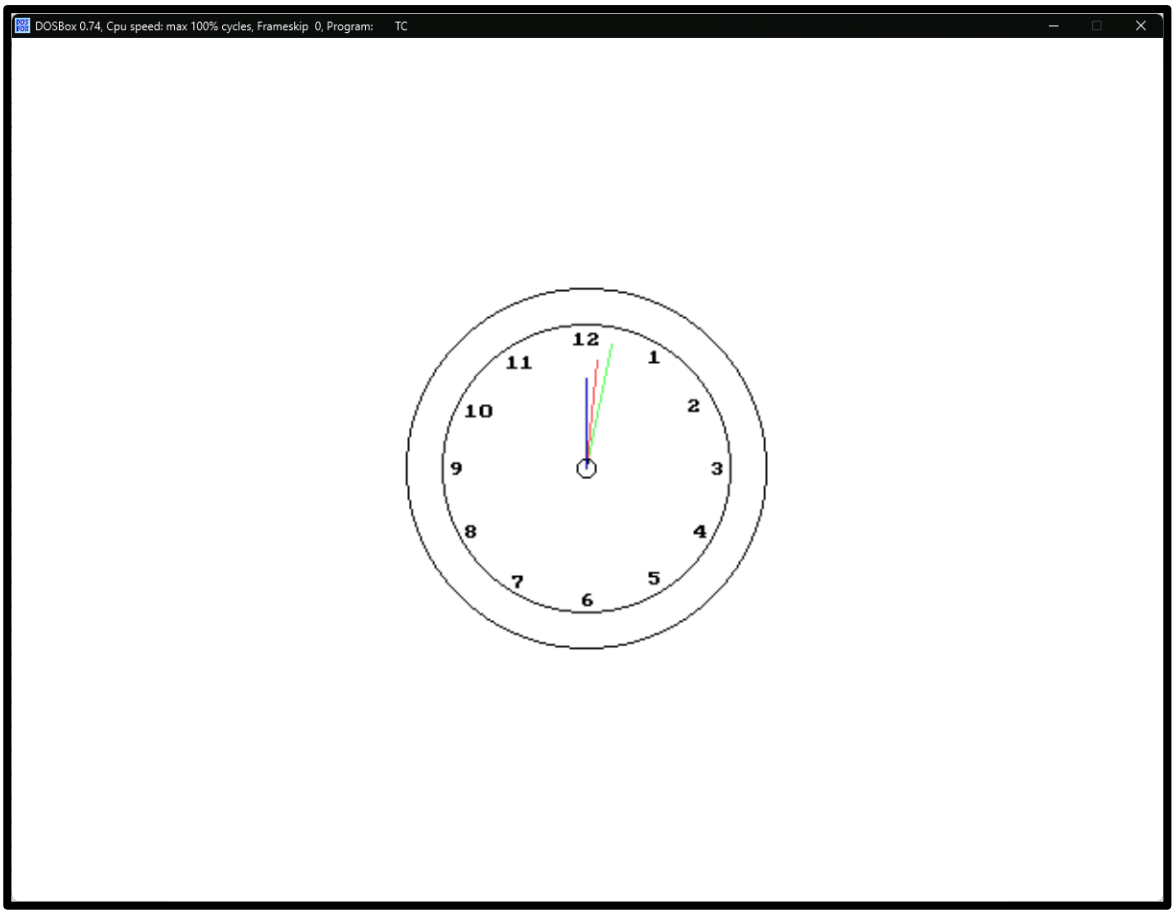




Q4. WAP to make an analog clock.

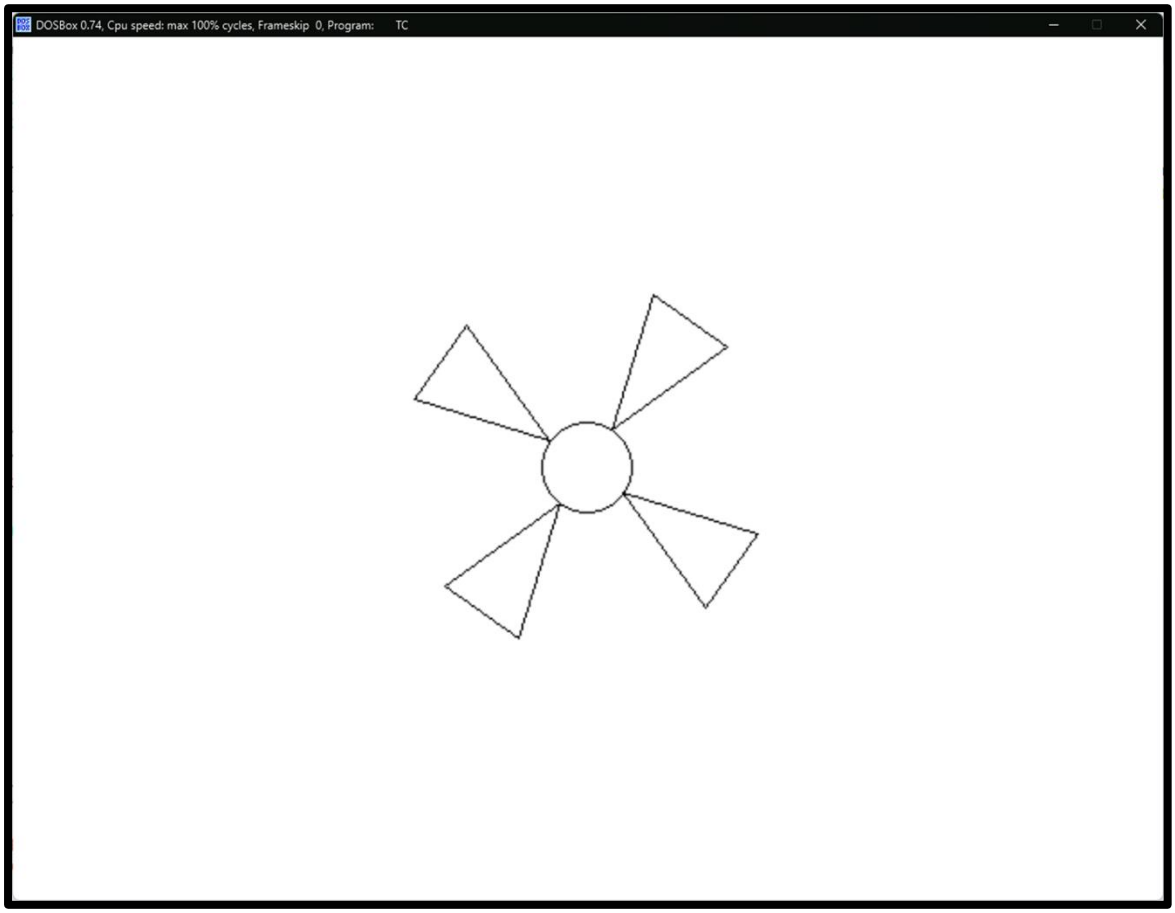
```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
#include<math.h>
#define pi 3.14f

float xSec=0.0f, ySec=-70.0f, xMin=0.0f, yMin=-60.0f, xHr=0.0f, yHr=-50.0f;
float a, b;
float angleSec=0.0f, angleMin=0.0f, angleHr=0.0f;

void layout() {
	circle(getmaxx()/2, getmaxy()/2, 100);
	circle(getmaxx()/2, getmaxy()/2, 80);
	outtextxy(getmaxx()/2-7, getmaxy()/2-75, "12");
	outtextxy(getmaxx()/2+35, getmaxy()/2-65, "1");
	outtextxy(getmaxx()/2+57, getmaxy()/2-38, "2");
	outtextxy(getmaxx()/2+70, getmaxy()/2-3, "3");
	outtextxy(getmaxx()/2+60, getmaxy()/2+32, "4");
	outtextxy(getmaxx()/2+35, getmaxy()/2+58, "5");
	outtextxy(getmaxx()/2-2, getmaxy()/2+70, "6");
	outtextxy(getmaxx()/2-41, getmaxy()/2+60, "7");
	outtextxy(getmaxx()/2-67, getmaxy()/2+32, "8");
	outtextxy(getmaxx()/2-75, getmaxy()/2-3, "9");
	outtextxy(getmaxx()/2-67, getmaxy()/2-35, "10");
	outtextxy(getmaxx()/2-44, getmaxy()/2-62, "11");
	circle(getmaxx()/2, getmaxy()/2, 5);
}

void hands(float x, float y, float angle, char col) {
	a = x*cos(angle) - y*sin(angle);
	b = x*sin(angle) + y*cos(angle);
	setcolor(WHITE);
	layout();
	if(col=='S')
		setcolor(GREEN);
	else if(col=='M')
		setcolor(RED);
	else if(col=='H')
		setcolor(BLUE);
	line(getmaxx()/2, getmaxy()/2, a+getmaxx()/2, b+getmaxy()/2);
}

void main() {
	int i, j;
	int gd = DETECT, gm;
	initgraph(&gd, &gm, "C:\\TC\\BGI");
	setbkcolor(WHITE);
	layout();
	while(1) {
		for(j=0 ; j<60 ; j++) {
			for(i=0 ; i<60 ; i++) {
				cleardevice();
				hands(xSec, ySec, angleSec, 'S');
				angleSec += (pi/30.0f);
				if(angleSec >= pi*2.0f) {
					angleSec -= pi*2.0f;
				}
				hands(xMin, yMin, angleMin, 'M');
				hands(xHr, yHr, angleHr, 'H');
```

```
                        delay(1000);
                }
                angleMin += (pi/30.0f);
                if(angleMin >= pi*2.0f) {
                        angleMin -= pi*2.0f;
                }
        }
        angleHr += (pi/30.0f);
        if(angleHr >= pi*2.0f) {
                angleHr -= pi*2.0f;
        }
    }
    getch();
    closegraph();
}
```



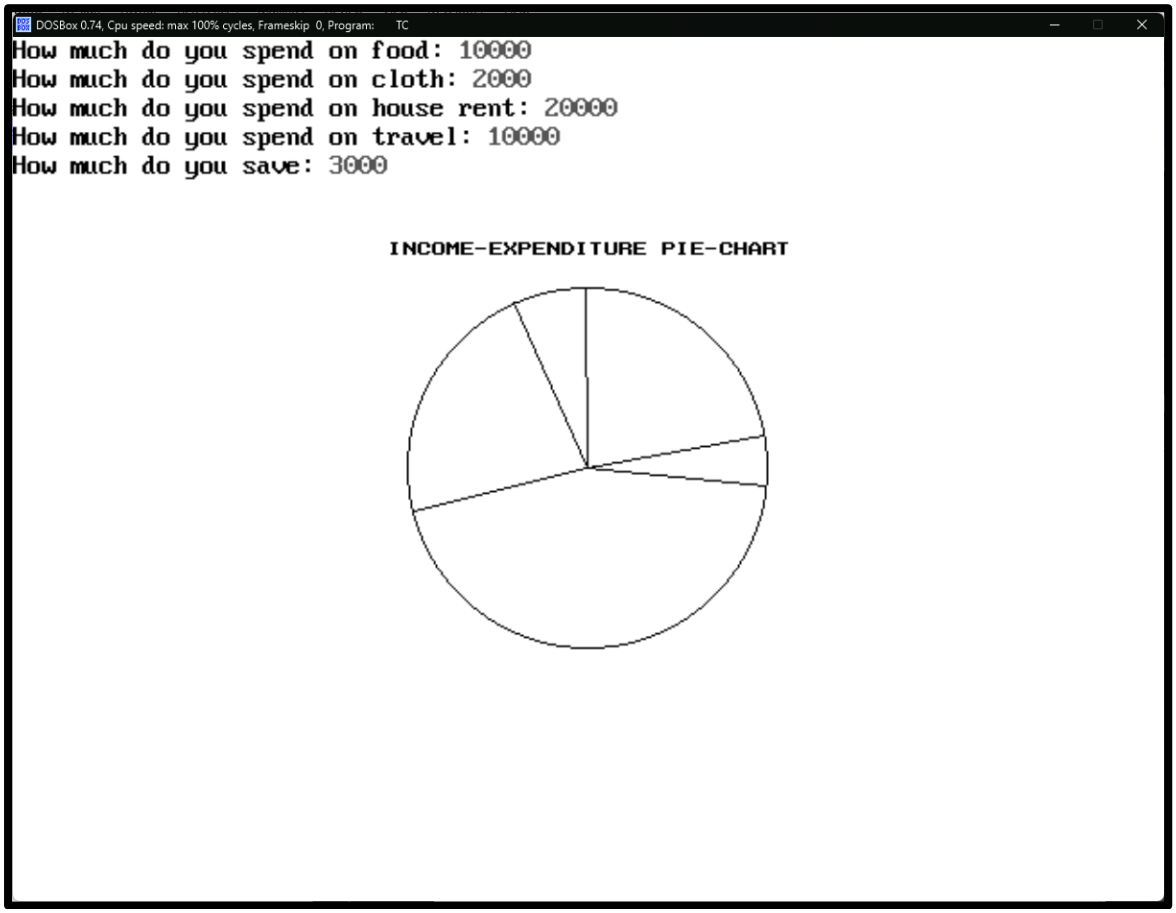
Q5. WAP to make a moving fan.

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
#include<math.h>
#define pi 3.14

float xa1=0.0f, xa2=-25.0f, xa3=25.0f, xb1=25.0f, xb2=100.0f, xb3=100.0f, xc1=0.0f,
xc2=25.0f, xc3=-25.0f, xd1=-25.0f, xd2=-100.0f, xd3=-100.0f;

float ya1=-25.0f, ya2=-100.0f, ya3=-100.0f, yb1=0.0f, yb2=-25.0f, yb3=25.0f, yc1=25.0f,
yc2=100.0f, yc3=100.0f, yd1=0.0f, yd2=25.0f, yd3=-25.0f;

float angle=0.0f, a1, b1, a2, b2, a3, b3;

void fans(float x1, float y1, float x2, float y2, float x3, float y3) {
      setcolor(BLACK);
      line(x1, y1, x2, y2);
      line(x2, y2, x3, y3);
      line(x3, y3, x1, y1);
}

void rotate(float x1, float y1, float x2, float y2, float x3, float y3, float angle) {
      a1 = x1*cos(angle) - y1*sin(angle);    b1 = x1*sin(angle) + y1*cos(angle);
      a2 = x2*cos(angle) - y2*sin(angle);    b2 = x2*sin(angle) + y2*cos(angle);
      a3 = x3*cos(angle) - y3*sin(angle);    b3 = x3*sin(angle) + y3*cos(angle);
      x1 = a1;     y1 = b1;
      x2 = a2;     y2 = b2;
      x3 = a3;     y3 = b3;
      fans(a1+getmaxx()/2, b1+getmaxy()/2, a2+getmaxx()/2, b2+getmaxy()/2,
a3+getmaxx()/2, b3+getmaxy()/2);
}

void main() {
      int i=0;
      int gd=DETECT, gm;
      initgraph(&gd, &gm, "C:\\TC\\BGI");
      setbkcolor(WHITE);
      while(1) {
            delay(10);
            cleardevice();
            rotate(xa1, ya1, xa2, ya2, xa3, ya3, angle);
            rotate(xb1, yb1, xb2, yb2, xb3, yb3, angle);
            rotate(xc1, yc1, xc2, yc2, xc3, yc3, angle);
            rotate(xd1, yd1, xd2, yd2, xd3, yd3, angle);
            angle += (pi/30.0f);
            circle(getmaxx()/2, getmaxy()/2, 25);
      }
      getch();
      closegraph();
}
```


DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC

Q6. WAP to draw a pie chart of family income and expenditure.

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
#include<math.h>
#define pi 3.14

float income, expd=0.0f, a, b;

void pieLine(float Per) {
	Per *= pi/50.0f;
	a = 100*sin(Per);
	b = -100*cos(Per);
	line(getmaxx()/2, getmaxy()/2, a + getmaxx()/2, b + getmaxy()/2);
}

void main() {
	float food, cloth, house, travel, save;

	int gd=DETECT, gm;
	initgraph(&gd, &gm, "C:\\TC\\BGI");
	setbkcolor(WHITE);
	setcolor(BLACK);

	printf("How much do you spend on food: ");
	scanf("%f", &food);
	printf("How much do you spend on cloth: ");
	scanf("%f", &cloth);
	printf("How much do you spend on house rent: ");
	scanf("%f", &house);
	printf("How much do you spend on travel: ");
	scanf("%f", &travel);
	printf("How much do you save: ");
	scanf("%f", &save);

	income = food + cloth + house + travel + save;
	food = (food*100.0f)/income;
	cloth = (cloth*100.0f)/income;
	house = (house*100.0f)/income;
	travel = (travel*100.0f)/income;
	save = (save*100.0f)/income;

	circle(getmaxx()/2, getmaxy()/2, 100);
	expd += food;	pieLine(expd);
	expd += cloth;	pieLine(expd);
	expd += house;	pieLine(expd);
	expd += travel;	pieLine(expd);
	expd += save;	pieLine(expd);

	outtextxy(getmaxx()/2 - 110, getmaxy()/2 - 125, "INCOME-EXPENDITURE PIE-CHART");

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


DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC
How much do you spend on food: 10000
How much do you spend on cloth: 2000
How much do you spend on house rent: 20000
How much do you spend on travel: 10000
How much do you save: 3000
INCOME-EXPENDITURE PIE-CHART