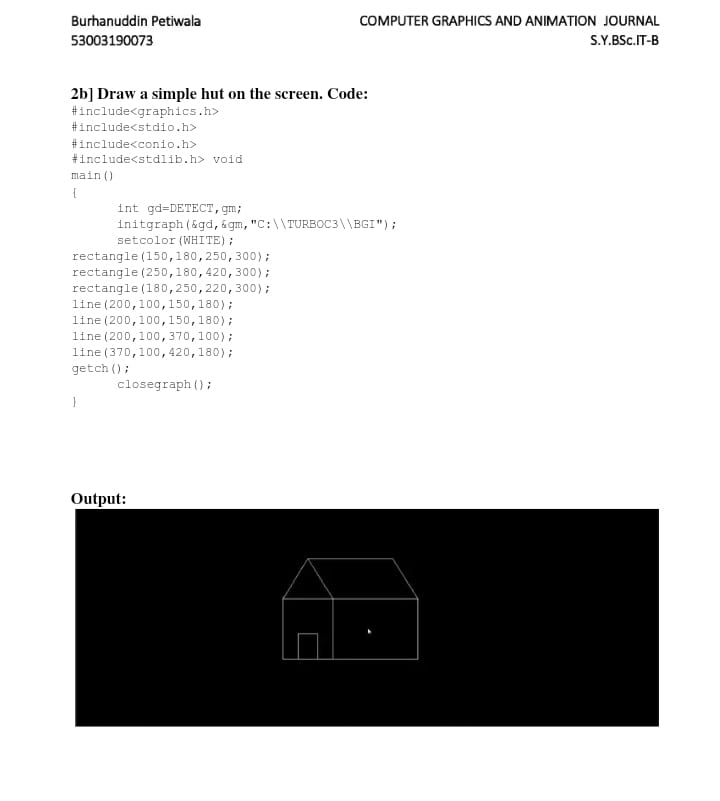
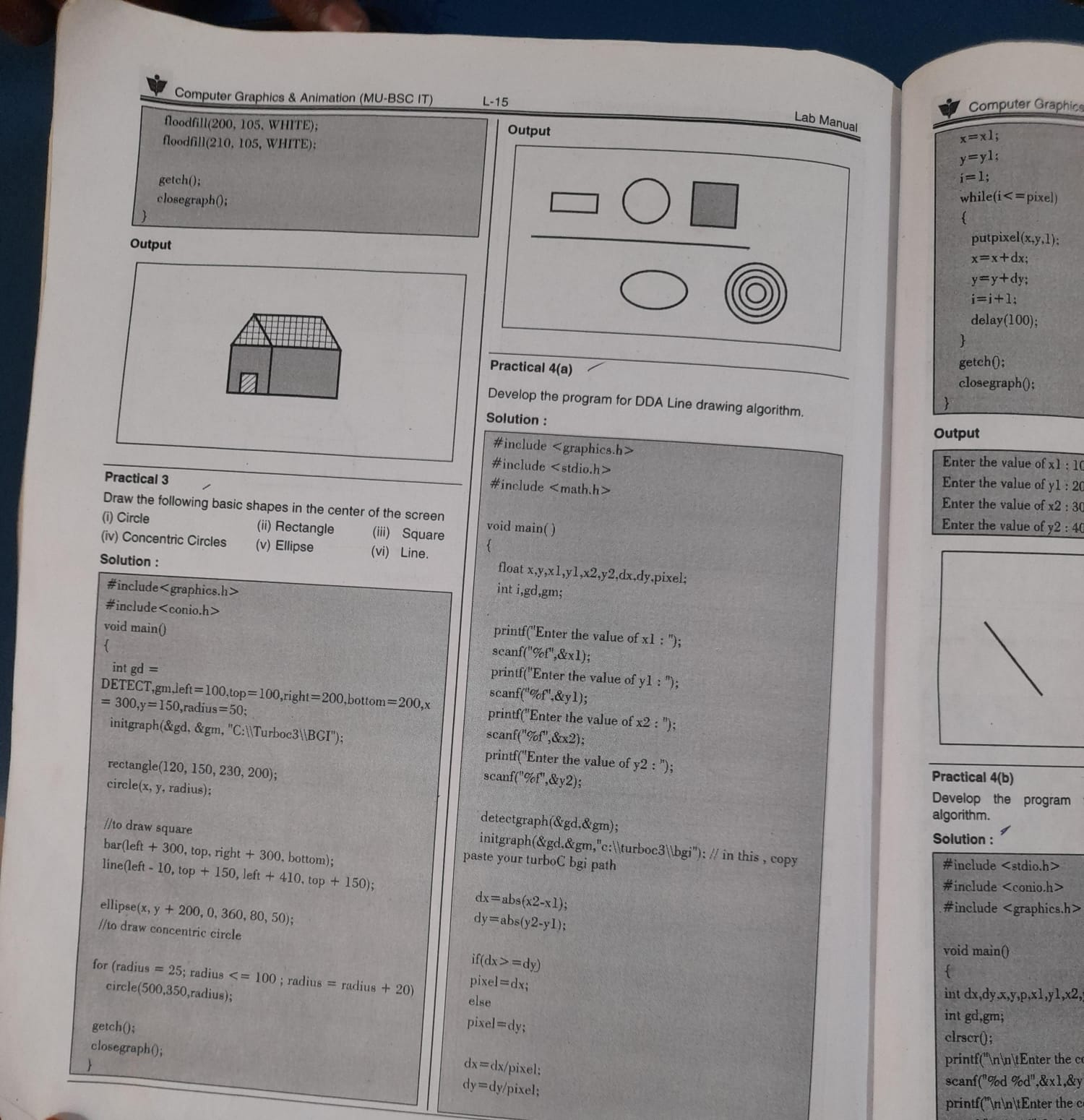
# 









**Practical – 4(A)**

**Aim:** Develop the program for DDA Line drawing algorithm.

**Source Code:**

#include<graphics.h>

#include<stdio.h>

#include<conio.h>

#include<math.h> #include<dos.h> void main()

{

float x,y,x1,y1,x2,y2,dx,dy,step; int i,gd=DETECT,gm; initgraph(&gd,&gm,"C:\\TC\\BGI"); printf("enter the value of x1 and y1"); scanf("%f%f",&x1,&y1);

printf("enter the value of x2 and y2"); scanf("%f%f",&x2,&y2); dx=abs(x2-x1); dy=abs(y2-y1); if(dx>=dy) step=dx; else step=dy; dx=dx/step; dy=dy/step; x=x1;

y=y1; i=1; while(i<=step)

{

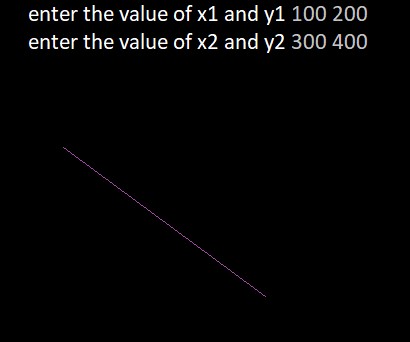
putpixel(x,y,5); x=x+dx; y=y+dy; i=i+1; delay(100);

}

closegraph(); getch();

}

**Output:**



# Practical – 4(B)

**Aim:** Develop the program for Bresenham’s Line drawing algorithm.

**Source Code:**

#include<stdio.h>

#include<conio.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;

}

}

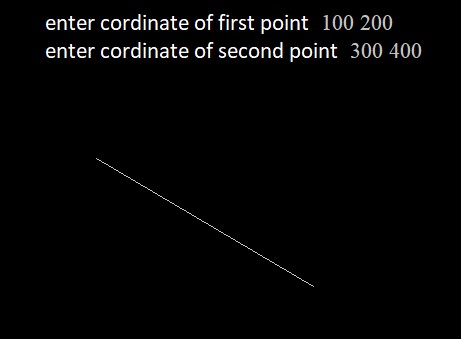
void main()

{

int gd=DETECT,gm,error,x0,y0,x1,y1; initgraph(&gd,&gm,"C:\\TC\\BGI"); printf("enter cordinate of first point"); scanf("%d%d",&x0,&y0); printf("enter cordinate of second point"); scanf("%d%d",&x1,&y1); drawline(x0,y0,x1,y1); getch(); closegraph();

}

**Output:**



## Practical – 5 (A)

**Aim:** Develop the program for the mid-point circle drawing algorithm.

**Source Code:**

#include<stdio.h>

#include<conio.h>

#include<graphics.h> #include<dos.h>

void drawcircle(int x0,int y0,int radius)

{

int x=radius; int y=0; int err=0; while(x>=y)

{

putpixel(x0+x,y0+y,7); putpixel(x0+y,y0+x,7); putpixel(x0-y,y0+x,7); putpixel(x0-x,y0+y,7); putpixel(x0-x,y0-y,7); putpixel(x0-y,y0-x,7); putpixel(x0+y,y0-x,7); putpixel(x0+x,y0-y,7); if(err<=0)

{

y+=1; err+=2\*y+1; }

if(err>=0)

{

x-=1; err-=2\*x+1;

}

}

}

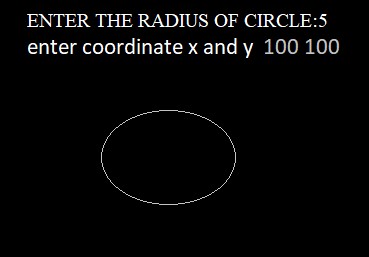
void main()

{

int gd=DETECT,gm,error,x,y,r; initgraph(&gd,&gm,"C:\\TC\\BGI"); printf("ENTER THE RADIUS OF CIRCLE:"); printf("enter coordinate x and y"); scanf("%d",&r); scanf("%d%d",&x,&y); drawcircle(x,y,r); closegraph(); getch();

}

**Output:**



## 6 (A

Write a program to perform 2D translation**.**

**Source Code:**

#include<graphics.h>

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

{

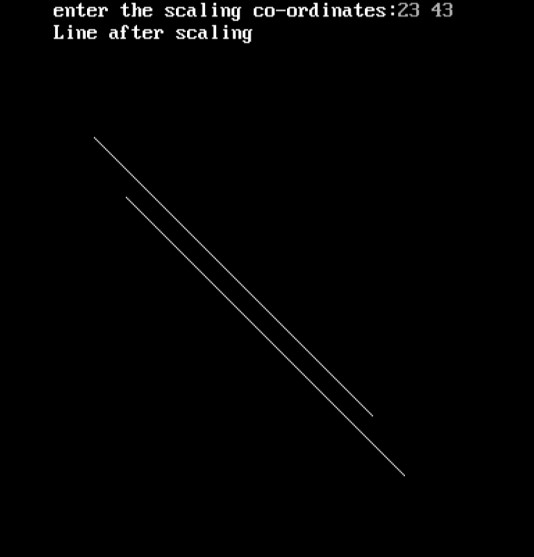
int graphdriver=DETECT,graphicmode; int x1,y1,x2,y2,x,y,x3,y3,x4,y4; printf("Enter the 2 lines end points:") ; printf("x1,y1,x2,y2"); scanf("%d%d%d%d",&x1,&y1,&x2,&y2); initgraph(&graphdriver,&graphicmode,"C:\\TC\\BGI"); line(x1,y1,x2,y2) ;

printf("enter the scaling co-ordinates:"); scanf("%d%d",&x,&y); x3=x1+x; y3=y1+y; x4=x2+x; y4=y2+y;

printf("Line after scaling"); line(x3,y3,x4,y4) ; getch(); closegraph();

}

**Output:**



## 6 (B

Write a program to implement 2D scaling**.**

**Source Code:**

#include<graphics.h>

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

{

int graphdriver=DETECT,graphicmode; int x1,y1,x2,y2,x,y,x3,y3,x4,y4; printf("Enter the 2 lines end points:") ; printf("x1,y1,x2,y2"); scanf("%d%d%d%d",&x1,&y1,&x2,&y2); initgraph(&graphdriver,&graphicmode,"C:\\TC\\BGI"); line(x1,y1,x2,y2) ;

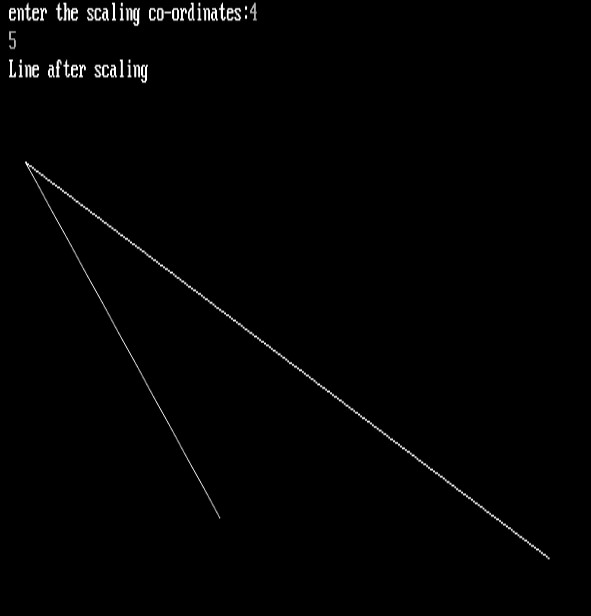
printf("enter the scaling co-ordinates:"); scanf("%d%d",&x,&y); x3=x1\*x; y3=y1\*y;

//x4=x2\*x; //y4=y2\*y;

printf("Line after scaling"); line(x1,y1,x4,y4) ; getch(); closegraph();

}

**Output:**



## 6 (C

Perform 2D Rotation on a given object**.**

**Source Code:**

#include<graphics.h>

#include<stdlib.h>

#include<stdio.h>

#include<math.h> #include<conio.h> void main()

{

int graphdriver = DETECT,graphmode; int i;

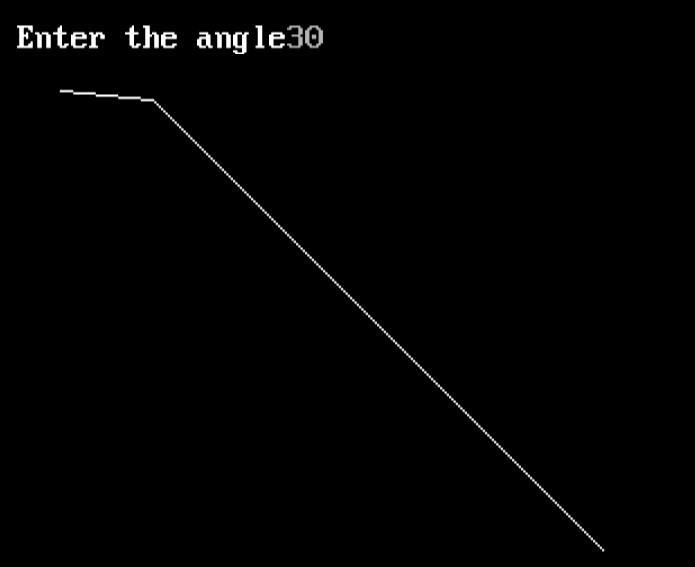
int x2,y2,x1,y1,x,y,xn,yn; double r11,r12,r21,r22,th; clrscr();

printf("Enter the 2 lie end points:"); printf("x1,y1,x2,y2"); scanf("%d%d%d%d",&x1,&y1,&x2,&y2); initgraph(&graphdriver,&graphmode,"C:\\TC\\BGI"); line(x1,y1,x2,y2); printf("\n\n\nEnter the angle"); scanf("%lf",&th); r11=cos((th\*3.1428)/180); r12=sin((th\*3.1428)/180); r21=(-sin((th\*3.1428)/180)); r22=cos((th\*3.1428)/180);

xn=((x2\*r11)-(y2\*r12)); yn=((x2\*r21)+(y2\*r22)); line(x1,y1,xn,yn); getch();

}

**Output:**



**Practical – 8 (B)**

Perform smiling face animation using graphic functions.

**Source Code:**

#include<graphics.h>

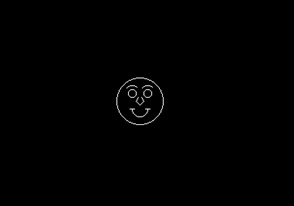
#include<conio.h> #include<stdio.h> void main()

{

int gd=DETECT,gm; initgraph(&gd,&gm,"C:\\TC\\BGI"); circle(200,200,30); circle(190,190,5); arc(190,190,50,130,10); circle(210,190,5); arc(210,190,50,130,10); arc(200,210,180,360,10); line(187,210,193,210); line(207,210,213,210); line(198,195,195,200); line(202,195,205,200); line(195,200,200,205); line(205,200,200,205); getch(); closegraph();

}

**Output:**



### Practical – 8 (C)

Draw the moving car on the screen.

**Source Code:**

#include<graphics.h>

#include<dos.h> #include<conio.h> void main()

{

int i,j=0,gd=DETECT,gm; initgraph(&gd,&gm,"C:\\TC\\BGI"); settextstyle(DEFAULT\_FONT,HORIZ\_DIR,2); outtextxy(25,240,"press any key to view the moving car"); getch();

setviewport(0,0,639,440,1); for(i=0;i<=420;i=i+10,j++)

{

rectangle(50+i,275,150+i,400); rectangle(150+i,350,200+i,400); circle(75+i,410,10); circle(175+i,410,10); setcolor(j); delay(100); if(i==420); break; clearviewport();

}

getch(); closegraph();

}

**Output:**

