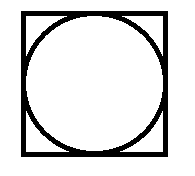
**֍ WRITE A PROGRAM TO DESIGN THE FOLLOWING DIAGRAM:**

****

PROGRAM CODE:

#include<stdio.h>

#include<conio.h>

#include<graphics.h>

int main()

{

clrscr();

int gd=DETECT,gm;

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

setlinestyle(SOLID\_LINE,0,THICK\_WIDTH);

setfillstyle(LTSLASH\_FILL,RED);

rectangle(10,10,100,100);

circle(55,55,10);

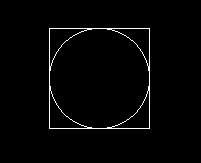
getch();

closegraph();

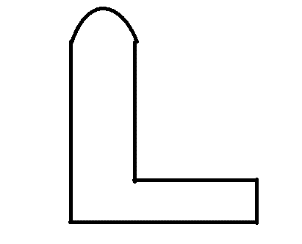
return 0;

}

OUTPUT:



**֍ WRITE A PROGRAM TO DESIGN THE FOLLOWING DIAGRAM AND SET COLOUR WTH BACKGROUND COLOUR:**



PROGRAM CODE:

#include<stdio.h>

#include<conio.h>

#include<graphics.h>

int main()

{

int gdrive=DETECT,gmode;

initgraph(&gdrive,&gmode,"c:\\turboc3\\bgi");

setbkcolor(GREEN);

setcolor(RED);

moveto(140,80);

arc(110,80,0,180,30);

moveto(80,80);

lineto(80,240);

lineto(340,240);

lineto(340,180);

lineto(140,180);

lineto(140,80);

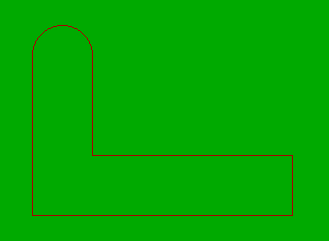
getch();

closegraph();

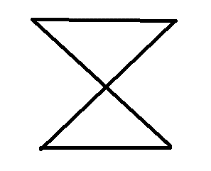
return(0);

}

OUTPUT:



**֍ WRITE A PROGRAM TO DESIGN THE FOLLOWING DIAGRAM USING DDA ALGORITHM.**

****

PROGRAM CODE:

#include<stdio.h>

#include<conio.h>

#include<graphics.h>

#include<math.h>

void dda(int,int,int,int);

int main()

{

int gd=DETECT,gm;

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

dda(100,100,200,100);

dda(100,100,200,200);

dda(200,100,100,200);

dda(100,200,200,200);

getch();

closegraph();

return 0;

}

void dda(int x1,int y1,int x2,int y2)

{

int dx,dy,step,i;

float x,y,xinc,yinc;

dx=x2-x1;

dy=y2-y1;

if(abs(dx)>abs(dy))

step=dx;

else

step=dy;

xinc=(float)dx/step;

yinc=(float)dy/step;

x=x1;

y=y1;

putpixel(x,y,WHITE);

for(i=1;i<=step;i++)

{

x=x+xinc;

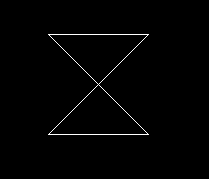
y=y+yinc;

putpixel(floor(x),floor(y),WHITE);

}

}

OUTPUT:



**֍ WRITE A PROGRAM TO DRAW A LINE USING BRESANHAM LINE DRAWING ALGORITHM.**

PROGRAM CODE:

#include<stdio.h>

#include<conio.h>

#include<graphics.h>

#include<math.h>

void bresen\_line(int,int,int,int);

int main()

{

int x1,y1,x2,y2,gd=DETECT,gm;

clrscr();

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

printf("Enter the two ending points");

scanf("%d%d%d%d",&x1,&y1,&x2,&y2);

bresen\_line(x1,y1,x2,y2);

getch();

return 0;

}

void bresen\_line(int x1,int y1,int x2,int y2)

{

int dx,dy,dt,ds,d,x,y;

dx=x2-x1;

dy=y2-y1;

dt=2\*(dy-dx);

ds=2\*dy;

d=2\*dy-dx;

x=x1;

y=y1;

putpixel(x,y,GREEN);

while(x<x2)

{

x++;

if(d<0)

d=d+ds;

else

{

y++;

d=d+dt;

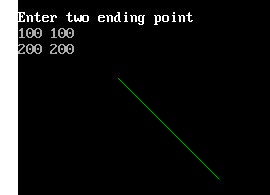
}

putpixel(x,y,GREEN);

}

}

OUTPUT:



**֍ DRAW A CIRCLE USING MID-POINT CIRCLE DRAWING ALGORITHM.**

PROGRAM CODE:

#include<stdio.h>

#include<conio.h>

#include<graphics.h>

void m\_circle(int,int,int);

void main()

{

int r,xc,yc,gd=DETECT,gm;

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

printf("Enter the radius");

scanf("%d",&r);

printf("Enter the center");

scanf("%d%d",&xc,&yc);

m\_circle(r,xc,yc);

getch();

}

void m\_circle(int r,int xc,int yc)

{

int p,x,y,gd=DETECT,gm;

p=1-r;

x=0;

y=r;

do

{

putpixel(xc+x,yc+y,6);

putpixel(xc+x,yc-y,6);

putpixel(xc-x,yc+y,6);

putpixel(xc-x,yc-y,6);

putpixel(xc+y,yc+x,6);

putpixel(xc+y,yc-x,6);

putpixel(xc-y,yc+x,6);

putpixel(xc-y,yc-x,6);

if(p>0)

{

p=p+2\*(x-y)+5;

y=y-1;

}

else

{

p=p+2\*x+3;

}

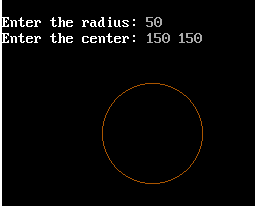
x=x+1;

}

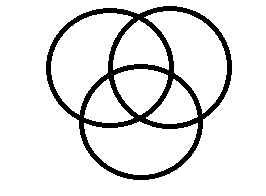
while(x<=y);

}

OUTPUT:



**֍ WRITE A PROGRAM TO DESIGN THE FOLLOWING DIAGRAM USING BRESENHAM CIRCLE DRAWING ALGORITHM.**



PROGRAM CODE:

#include<stdio.h>

#include<conio.h>

#include<graphics.h>

void b\_circle(int,int,int);

void main()

{

int r,xc,yc,gd=DETECT,gm;

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

b\_circle(40,150,150);

b\_circlr(40,180,150);

b\_circle(40,165,190);

getch();

}

void b\_circle(int r,int xc,int yc)

{

int p,x,y;

p=3-2\*r;

x=0;

y=r;

do

{

putpixel(xc+x,yc+y,6);

putpixel(xc+x,yc-y,6);

putpixel(xc-x,yc+y,6);

putpixel(xc-x,yc-y,6);

putpixel(xc+y,yc+x,6);

putpixel(xc+y,yc-x,6);

putpixel(xc-y,yc+x,6);

putpixel(xc-y,yc-x,6);

if(p>0)

{

p=p+4\*(x-y)+10;

y=y-1;

}

else

{

p=p+4\*x+6;

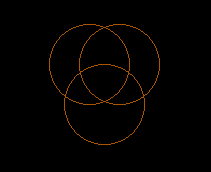
}

x=x+1;

}while(x<=y);

}

OUTPUT:



**֍ WRITE A PROGRAM TO SCALE A SQUARE, TWO UNIT IN X DIRECTION AND TWO UNIT IN Y DIRECTION WITH RESPECT TO ORIGIN.**

PROGRAM CODE:

#include<stdio.h>

#include<conio.h>

#include<graphics.h>

#include<math.h>

int main()

{

int gd=DETECT,gm,m,f,x,y,i,j,k, n,s,a[10][10],

b[10][10],c[10][10];

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

line(0,239,639,239);

line(319,0,319,479);

m=getmaxx()/2;

f=getmaxy()/2;

printf("How many point");

scanf("%d",&n);

printf("Enter the points");

for(i=0;i<n;i++)

scanf("%d%d",&a[0][i],&a[1][i]);

for(i=0;i<n-1;i++)

line(m+a[0][i],f-a[1][i],m+a[0][i+1],

f-a[1][i+1]);

line(m+a[0][0],f-a[1][0],m+a[0][i],f-a[1][i]);

printf("Enter the scalling factor");

scanf("%d%d",&x,&y);

b[0][0]=x;

b[0][1]=0;

b[1][0]=0;

b[1][1]=y;

for(i=0;i<2;i++)

{

for(j=0;j<n;j++)

{

c[i][j]=0;

for(k=0;k<2;k++)

c[i][j]=c[i][j]+(b[i][k]\*a[k][j]);

}

}

for(i=0;i<n-1;i++)

line(m+c[0][i],f-c[1][i],m+c[0][i+1],

f-c[1][i+1]);

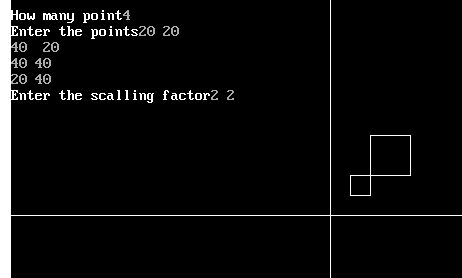
line(m+c[0][0],f-c[1][0],m+c[0][i],f-c[1][i]);

getch();

return 0;

}

OUTPUT:



**֍ WRITE A PROGRAM TO PERFOMR A 90̊ ROTATION OF A TRIANGLE.**

PROGRAM CODE:

#include<stdio.h>

#include<conio.h>

#include<math.h>

#include<graphics.h>

void main()

{

int gd=DETECT,gm;

float td,i,j,k,r,n,s,a[10][10],b[10][10],

c[10][10],m,f,t;

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

line(0,239,639,239);

line(319,0,319,479);

m=getmaxx()/2;

f=getmaxy()/2;

printf("How many number");

scanf("%f",&n);

printf("Enter the point");

for(i=0;i<n;i++)

scanf("%f%f",&a[0][i],&a[1][i]);

for(i=0;i<n-1;i++)

line(m+a[0][i],f-a[1][i],m+a[0][i+1],

f-a[1][i+1]);

line(m+a[0][0],f-a[1][0],m+a[0][i],f-a[1][i]);

printf("Enter the angle");

scanf("%f",&td);

t=(3.141\*td)/180;

b[0][0]=cos(t);

b[0][1]=-sin(t);

b[1][0]=sin(t);

b[1][1]=cos(t);

for(i=0;i<2;i++)

{

for(j=0;j<n;j++)

{

c[i][j]=0;

for(k=0;k<2;k++)

{

c[i][j]=c[i][j]+(b[i][k]\*a[k][j]);

}

}

}

for(i=0;i<n-1;i++)

line(m+c[0][i],f-c[1][i],m+c[0][i+1],

f-c[1][i+1]);

line(m+c[0][0],f-c[1][0],m+c[0][i],f-c[1][i]);

getch();

}

OUTPUT:

