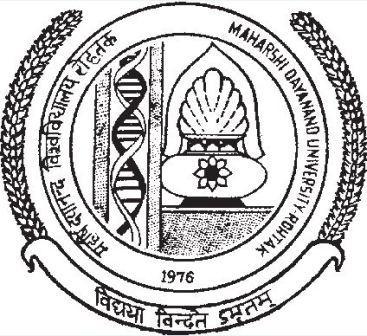
Department Of Computer Science & Application



Computer Graphics Using C



**INDEX**

| **S.No** | **Practical** | **Remarks** |
| --- | --- | --- |
| **1** | Write a program in C to print Random Pixel. |  |
| **2** | Write a program to implement person moving in rain. |  |
| **3** | Write a program to draw 3D bar on C. |  |
| **4** | Write a program to implement a bouncing ball in C. |  |
| **5** | Write a program to make a snake game in C. |  |
| **6** | Write a program to implement the DDA algorithm. |  |
| **7** | Write a program to implement Bresenham’s Line Drawing Algorithm. |  |
| **8** | Write a program to implement the mid-point algorithm. |  |
| **9** | Write a program to implement Bresenham’s circle drawing algorithm. |  |
| **10** | Write a program to implement translation in C |  |
| **11** | Write a program to implement scale image in C. |  |
| **12** | Write a program to implement shearing in C. |  |
| **13** | Write a program to implement Reflection in C. |  |
| **14** | Write a program to implement rotation in C. |  |
| **15** | Write a program to implement flood fill in C. |  |

**1 Write a program in C to print Random Pixel.**

#include<stdio.h>

#include<graphics.h>

#include<conio.h>

#include<stdlib.h>

int main(){

int gd= DETECT, gm;

int x,y,c,i;

initgraph(&gd,&gm,"c://TURBOC3//BGI");

cleardevice();

for(i=0;i<100;++i){

x=rand()%601;

y=rand()%501;

c=rand()%16;

putpixel(x,y,c);

}

getch();

closegraph();

}

**OUTPUT:**

****

**2 Write a program to implement person moving in rain.**

#include<stdio.h>

#include<graphics.h>

#include<conio.h>

#include<stdlib.h>

int main(){

int gd=DETECT,gm;

int x,y,i=0,j;

initgraph(&gd,&gm,"c://TURBOC3//BGI");

while(!kbhit()){

circle(20+i,384,14); //head

line(20+i,400,20+i,430); //chest

line(20+i,430,12+i,454); //leg left

line(20+i,430,28+i,454); //leg right

line(20+i,410,10+i,420); //left hand

line(20+i,410,30+i,420); //right hand

line(30+i,420,30+i,360); //umbral handle

line(0+i,360,60+i,360); //umbrella bottom;

arc(30+i,360,0,180,30); //umbrella above;

for(j=0;j<600;++j){

x=rand()%651;

y=rand()%451;

line(x-2,y+2,x,y);

}

delay(10);

cleardevice();++i;

}

getch();

closegraph();

}

**OUTPUT:**

****

****

****



**3 Write a program to draw 3D bar on C.**

#include<stdio.h>

#include<graphics.h>

#include<conio.h>

int main(){

int gd=DETECT,gm;

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

//bar

setfillstyle(5,3);

bar3d(100,100,120,420,5,1);

bar3d(140,200,160,420,5,1);

bar3d(180,240,200,420,5,1);

bar3d(220,170,240,420,5,1);

//line;

setlinestyle(2,0,2);

line(90,430,300,430);

line(90,430,90,40);

//text

settextstyle(1,0,3);

outtextxy(70,50,"X");

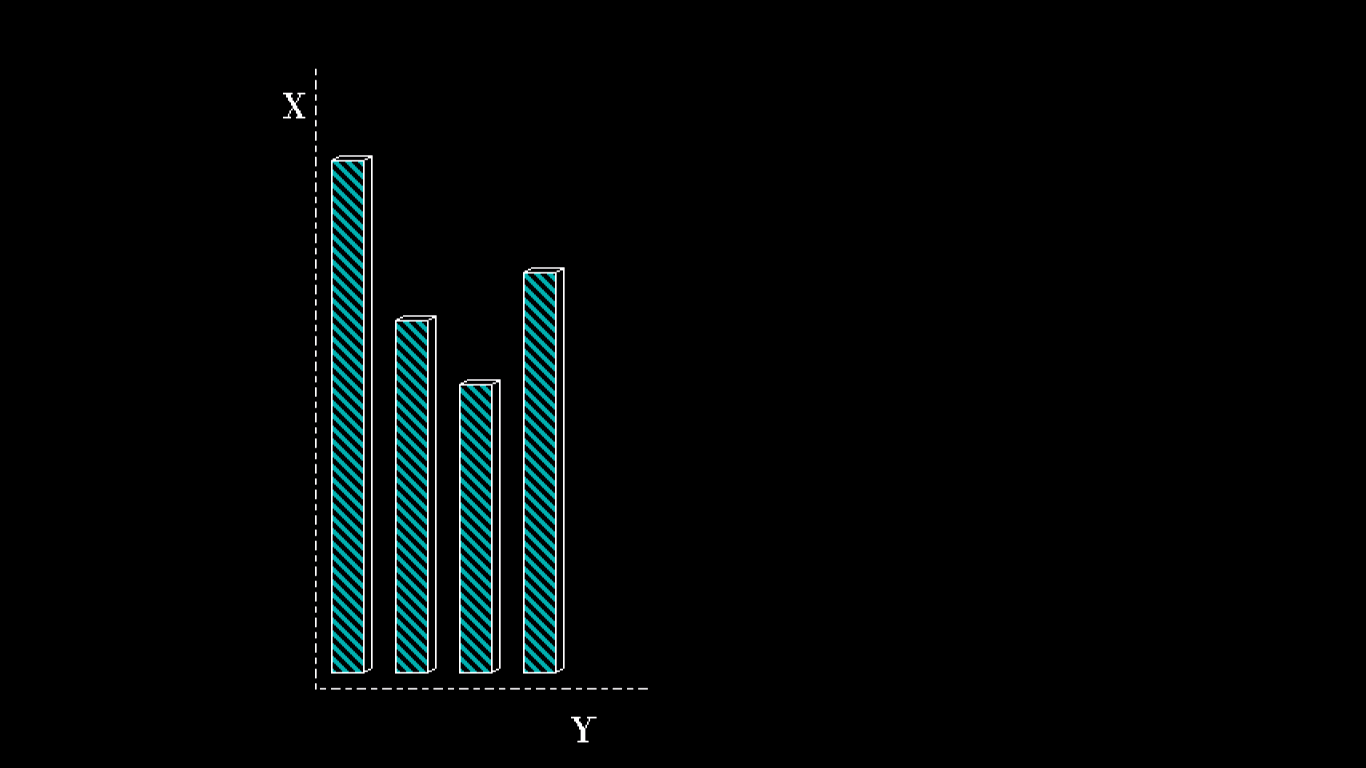
outtextxy(250,440,"Y");

getch();

closegrah();

}

**OUTPUT:**

****

**4 Write a program to implement a bouncing ball in C.**

#include<stdio.h>

#include<graphics.h>

#include<conio.h>

int main(){

int gd=DETECT,gm;

int x=0,y=0;

initgraph(&gd,&gm,"C://TURBOC3//BGI");

while(!kbhit()){

setfillstyle(8,BROWN);

line(0,410,650,410);

floodfill(12,12,WHITE);

circle(x+84,y+85,50);

delay(50);

cleardevice();

if(x<500 && y<260){

x=x+10;

y=y+15;

}

else{

x=x+5;

y=y-5;

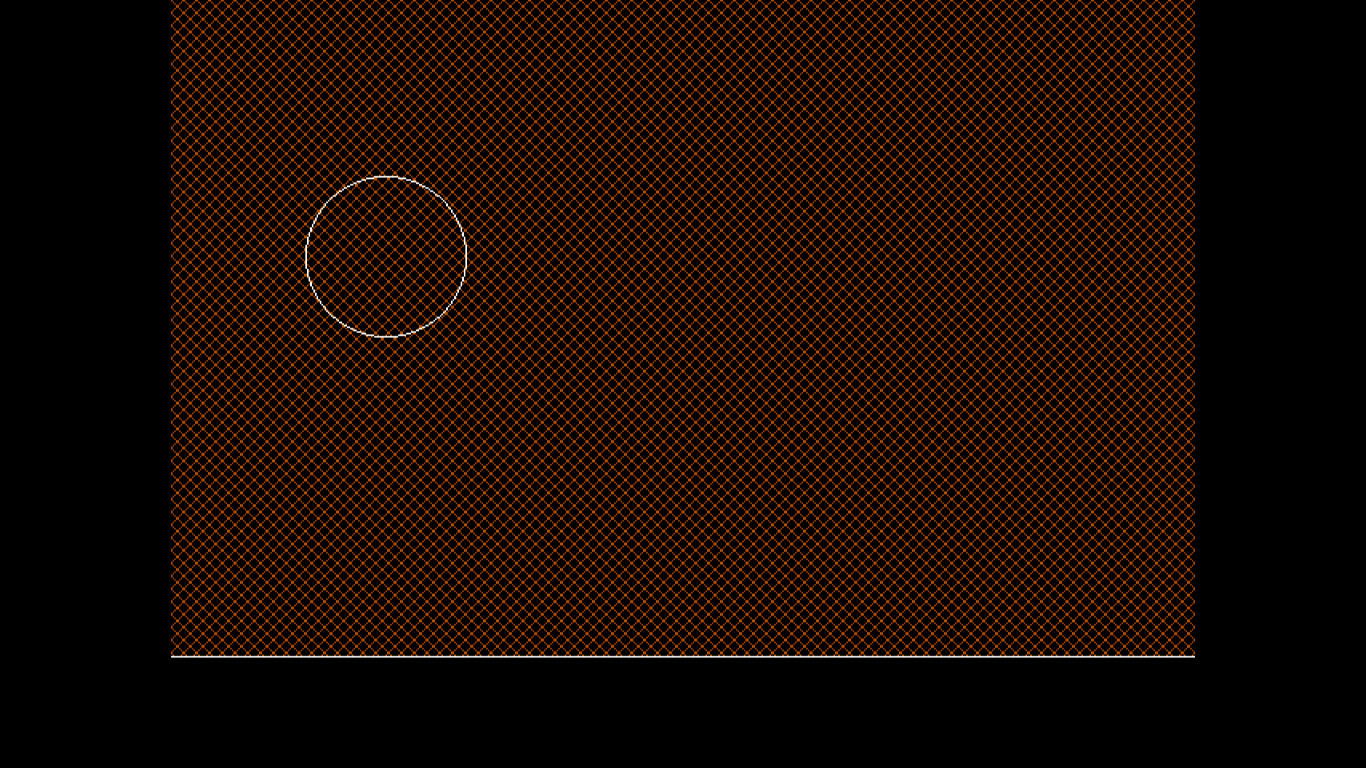
}

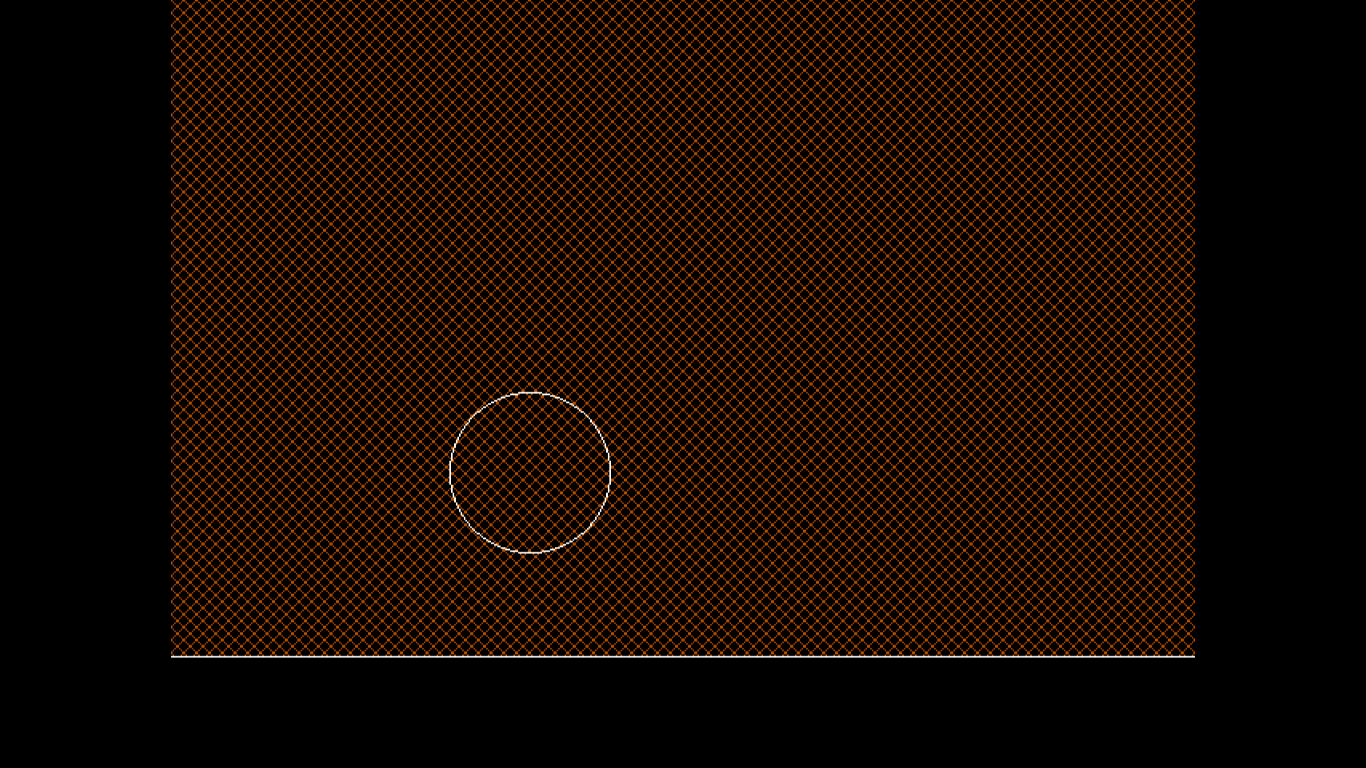
}

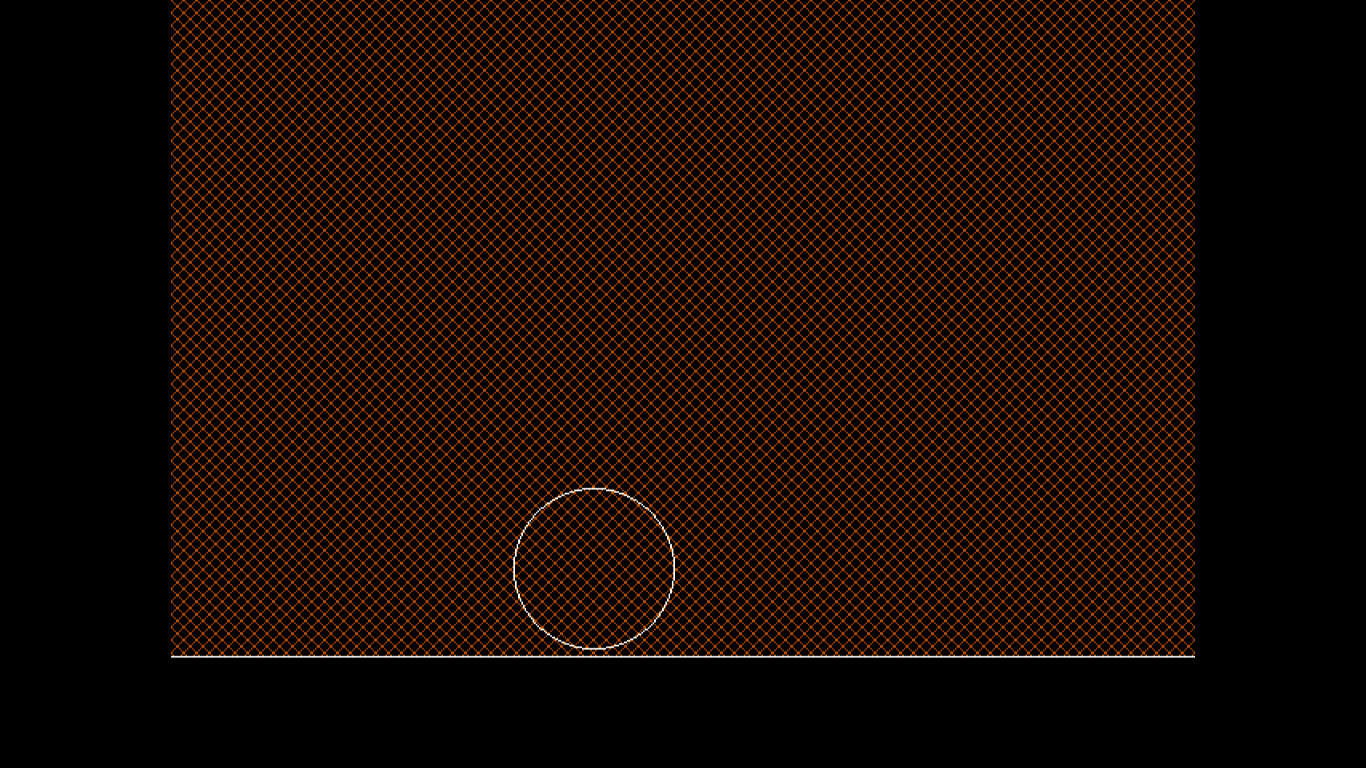
getch();

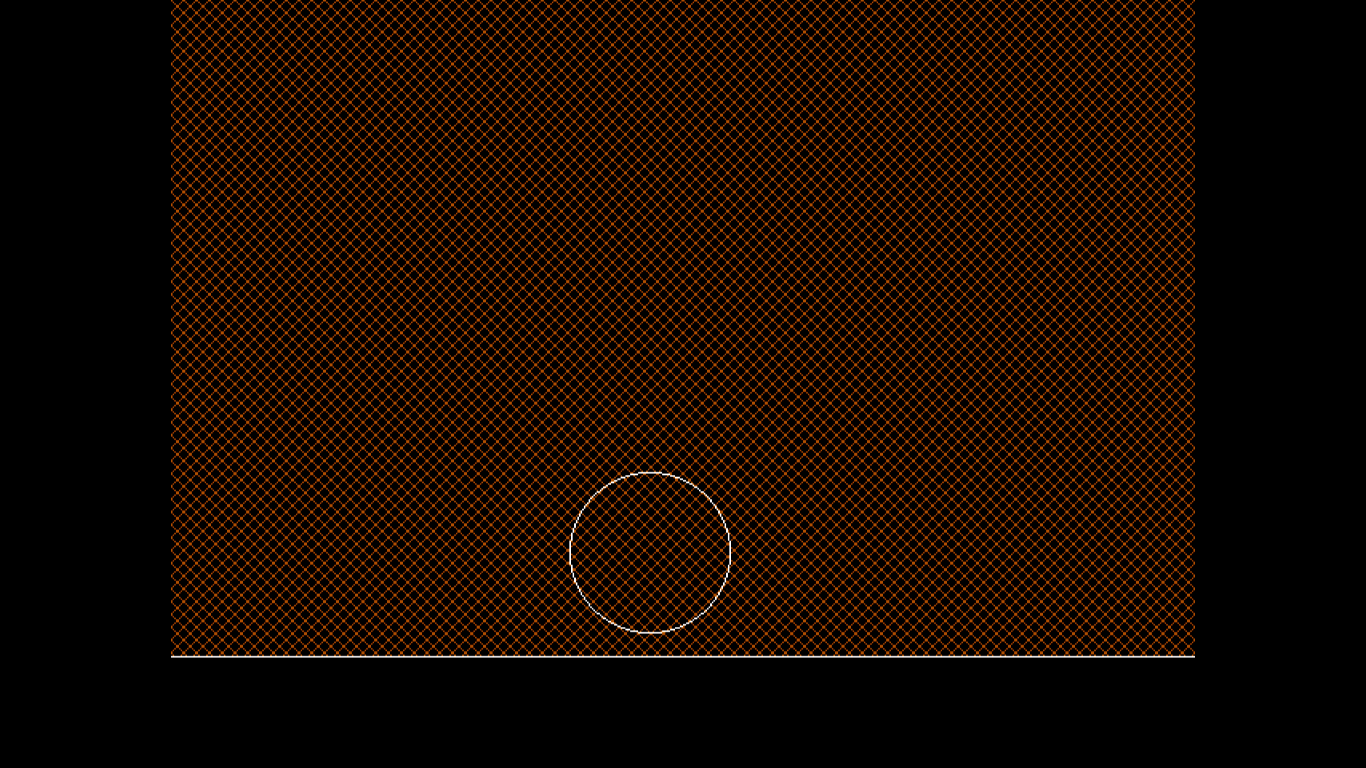
}

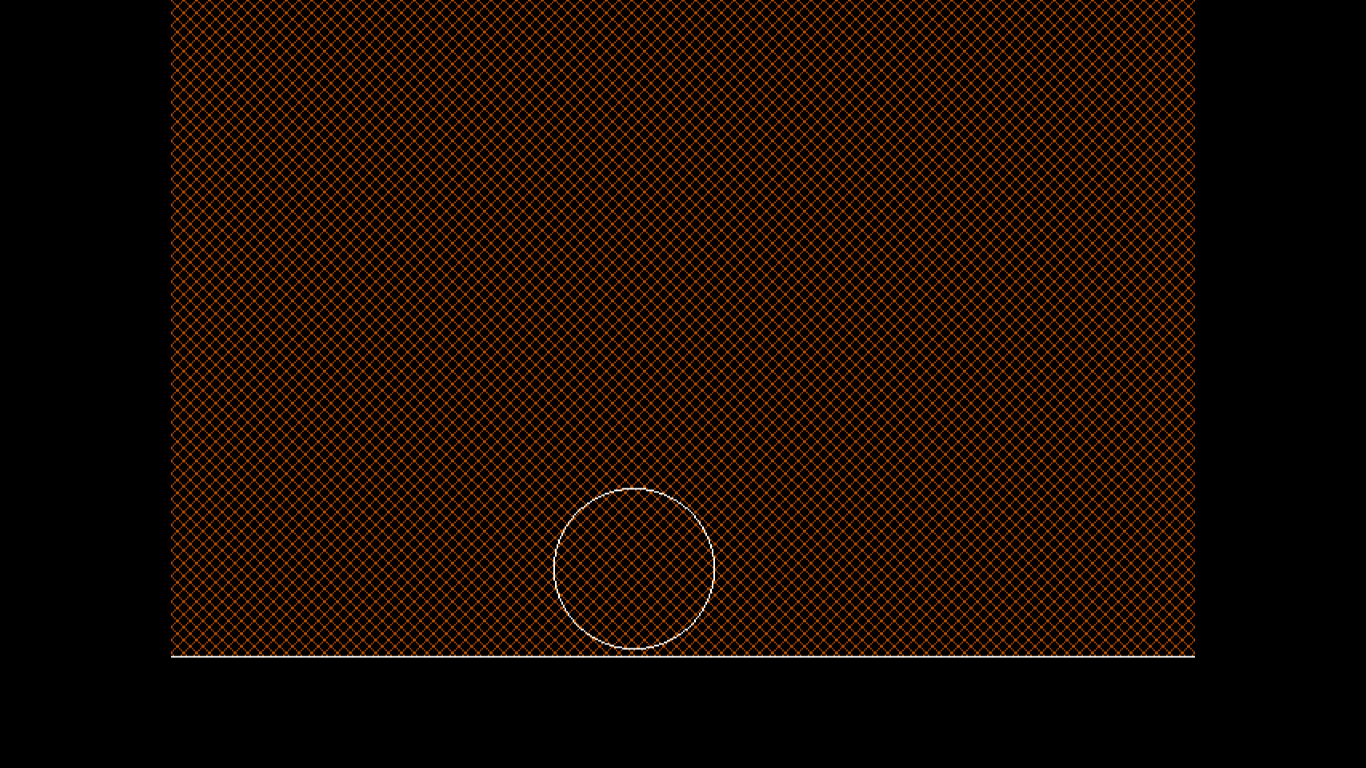
**OUTPUT:**

****

****

****

****

****

**5 Write a program to implement moving ceiling fan.**

#include<stdio.h>

#include<graphics.h>

#include<conio.h>

int converter(int co){

return 20+co\*10;

}

void putpixelBhumit(int x,int y){

rectangle(converter(x),converter(y),converter(x+1),converter(y+1));

}

int newXCord(){

return rand()%58+1;

}

int newYCord(){

return rand()%38+1;

}

int main(){

int x=22,y=22;

enum {LEFT,RIGHT,TOP,BOTTOM};

int status=LEFT;

int gd=DETECT,gm;

int score=0;

int randx,randy;

char ch;

initgraph(&gd,&gm,"C://TURBOC3//BGI");

randx=rand()%56+1;

randy=rand()%36+1;

while(x>=0 && x<58 && y>=0 && y<38){

while(!kbhit() && x>=0 && x<58 && y>=0 && y<38){

bar(0,0,20,400);

bar(600,0,620,400);

bar(0,400,620,420);

bar(0,0,620,20);

putpixelBhumit(randx,randy);

switch(status){

case LEFT: x=x-1;break ;

case RIGHT: x=x+1;break ;

case TOP: y=y-1;break ;

case BOTTOM: y=y+1;break;

}

delay(500);

cleardevice();

putpixelBhumit(x,y);

if(x==randx && y==randy){

randx=newXCord();

randy=newYCord();

score=score+1;

}

}

if(kbhit()){

ch=getch();

switch(ch){

case 'w':status=TOP;break;

case 's':status=BOTTOM;break;

case 'a':status=LEFT;break;

case 'd':status=RIGHT;break;

case 'p':{

return 0;

}

}

printf("%c %d\n",ch,status);

}

}

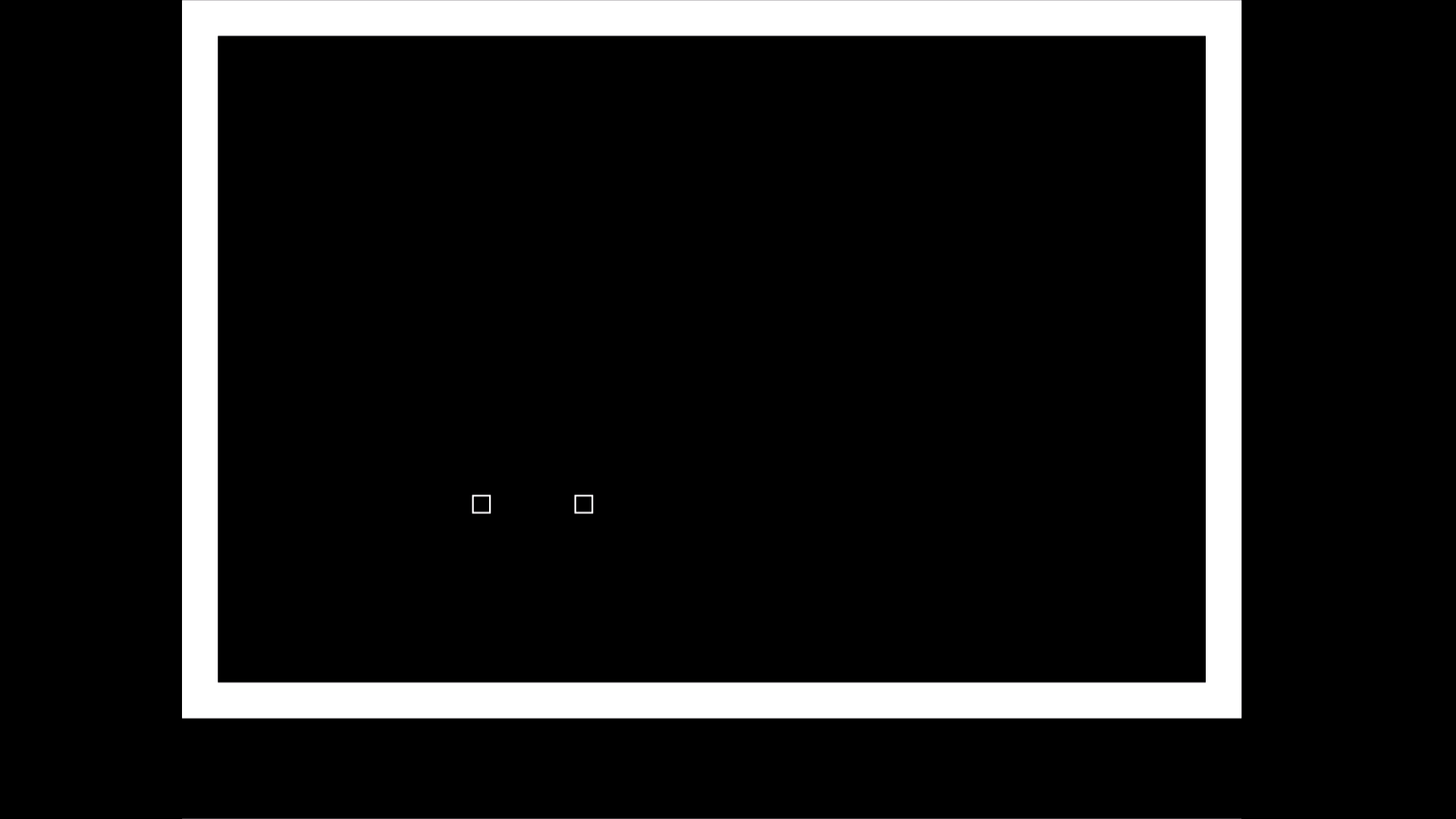
printf("Game Over:\t");

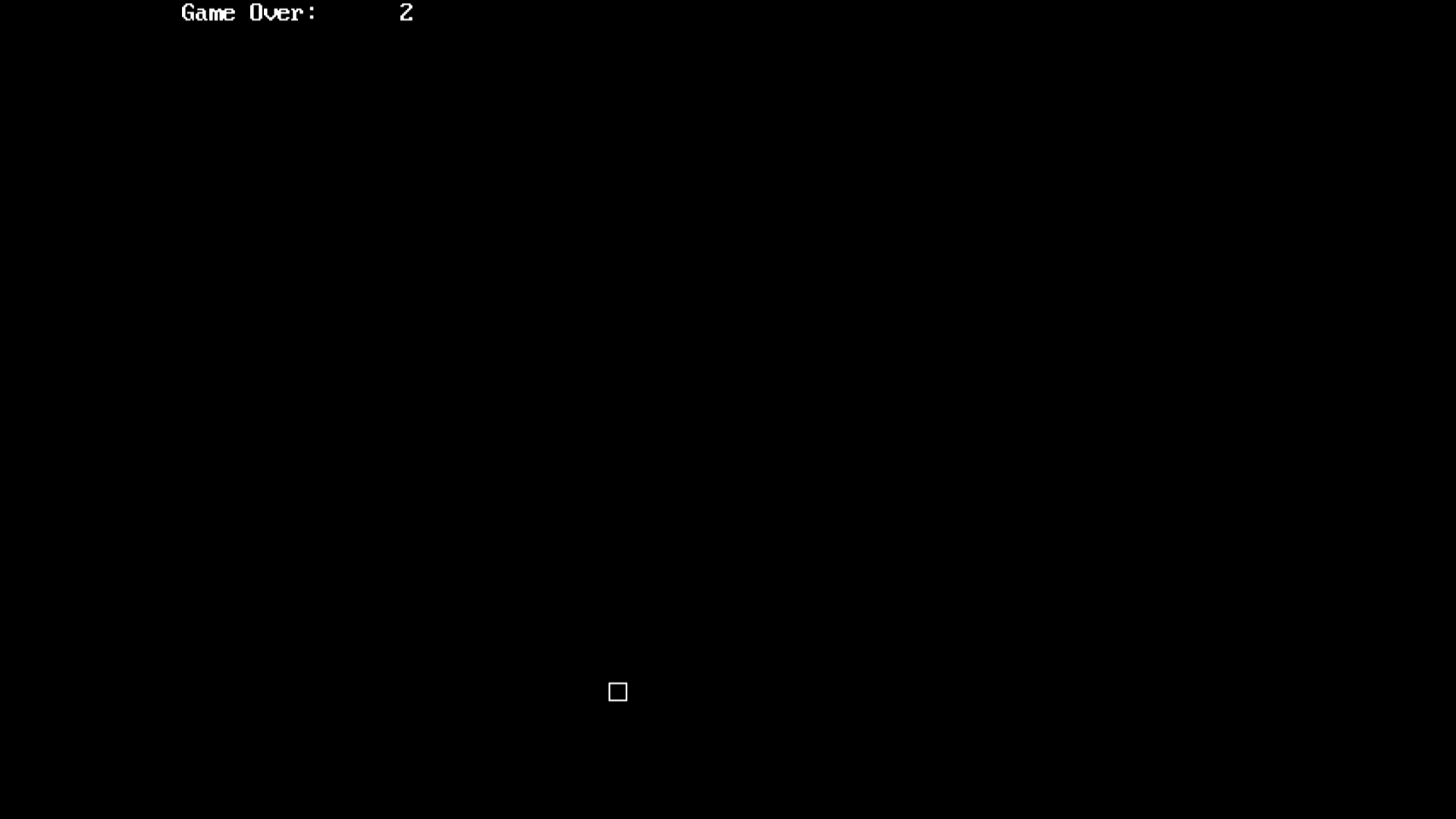
printf("%d",score);

getch();

}

**OUTPUT:**

****

****

**6 Write a program implementing the DDA algorithm.**

#include<stdio.h>

#include<graphics.h>

#include<conio.h>

#include<math.h>

int main(){

int gd=DETECT, gm;

float x1,y1,x2,y2;

float x,y,i;

float slope;

float dx,dy;

initgraph(&gd,&gm,"C://TURBOC3//BGI");

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

dx=x2-x1;

dy=y2-y1;

printf("%f %f", dx,dy);

if(dx>=dy) slope=dx;

else slope=dy;

dx=dx/slope;

dy=dy/slope;

x=x1;

y=y1;

i=1;

while(i<=slope){

putpixel((int)(x),(int)y,YELLOW);

x=x+dx;

y=y+dy;

i=i+1;

delay(10);

}

getch();

closegraph();

}

**OUTPUT:**

****

**7 Write a program to implement Bresenham’s Line Drawing Algorithm**

#include<stdio.h>

#include<graphics.h>

#include<conio.h>

int main(){

int x1,y1,x2,y2,dx,dy,x,y,endx,endy,p;

int gd=DETECT,gm;

initgraph(&gd,&gm,"C://TURBOC3//BGI");

printf("Enter coordinates of first point: ");

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

printf("\nEnter coordinates of second point: ");

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

dx=x2-x1;

dy=y2-y1;

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

if(x1>x2){

x=x2;

y=y2;

endx=x1;

}else{

x=x1;

y=y1;

endx=x2;

}

p=2\*dy-dx;

while(x<endx){

putpixel(x,y,WHITE);

if(p<0){

p=p+2\*dy;

}else{

y++;

p=p+2\*dy-2\*dx;

}

x++;

}

}

else{

if(y1>y2){

y=y2;

x=x2;

endy=y1;

}else{

x=x1;

y=y1;

endy=y2;

}

dx=abs(dx);

dy=abs(dy);

p=2\*dx-dy;

while(y<endy){

putpixel(x,y,RED);

if(p<0){

p=p+2\*dx;

}else{

x++;

p=p+2\*dx-2\*dy;

}

y++;

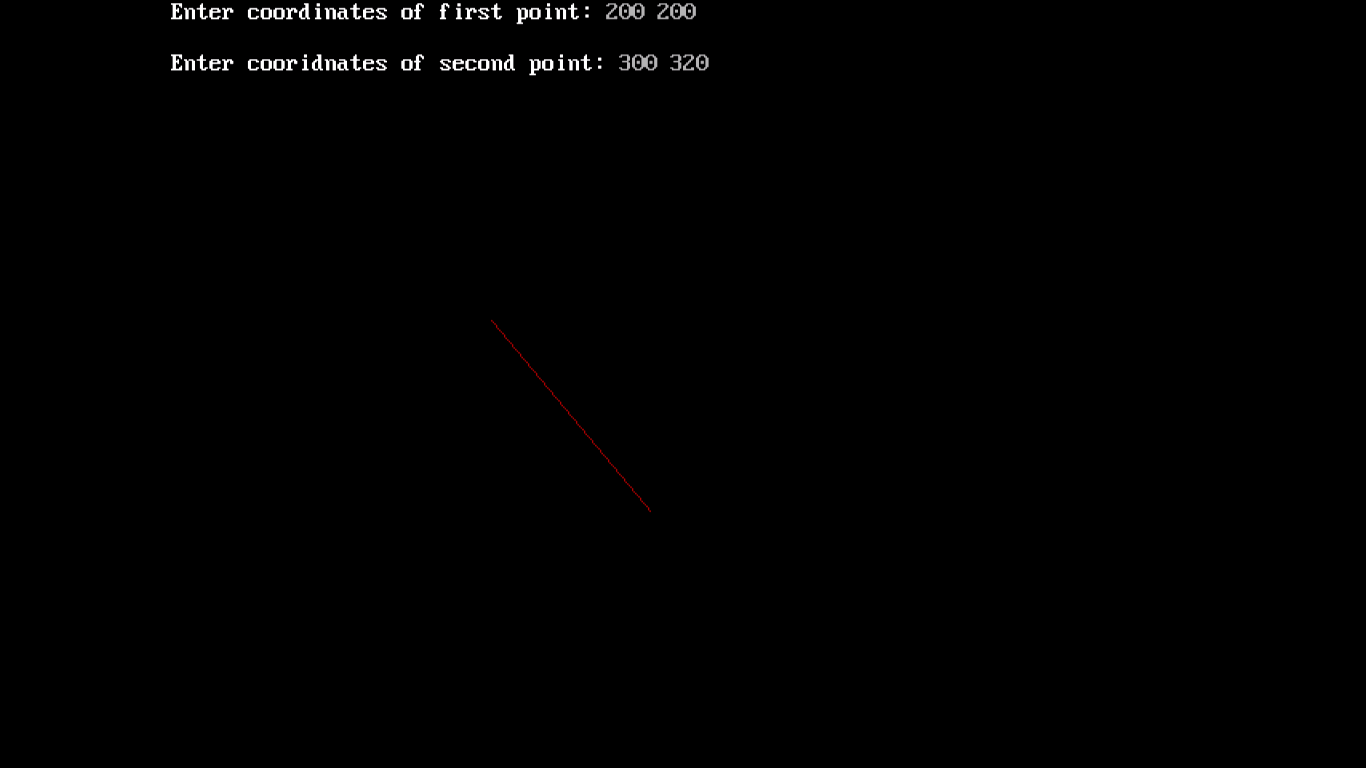
}

}

getch();

}

**OUTPUT:**



**8 Write a program implementing the mid-point algorithm.**

#include<stdio.h>

#include<graphics.h>

#include<conio.h>

int main(){

int gd=DETECT,gm;

int xc,yc,x=0,y=0,r,p;

initgraph(&gd,&gm,"C://TURBOC3//BGI");

printf("Enter centre and radius:\t");

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

p=1-r;

y=r;

cleardevice();

while(x<=y){

if(p<0){

x=x+1;

y=y;

p=p+2\*x+1;

}

else{

x=x+1;

y=y-1;

p=p+2\*x+1-2\*y;

}

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

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

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

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

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

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

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

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

}

getch();

closegraph();

}

**OUTPUT:**

****

**9 Write a program to implement Bresenham’s circle drawing algorithm.**

#include<stdio.h>

#include<graphics.h>

#include<conio.h>

int main(){

int gd=DETECT,gm;

int xc,yc,x,y,r;

int x,y,p;

initgraph(&gd,&gm,"C://TURBOC3//BGI");

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

x=0;

y=r;

p=3-2\*r;

while(x<=y){

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

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

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

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

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

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

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

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

if(p<0){

x=x+1;

p=p+4\*x+6;

}

else{

x=x+1;

y=y-1;

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

}

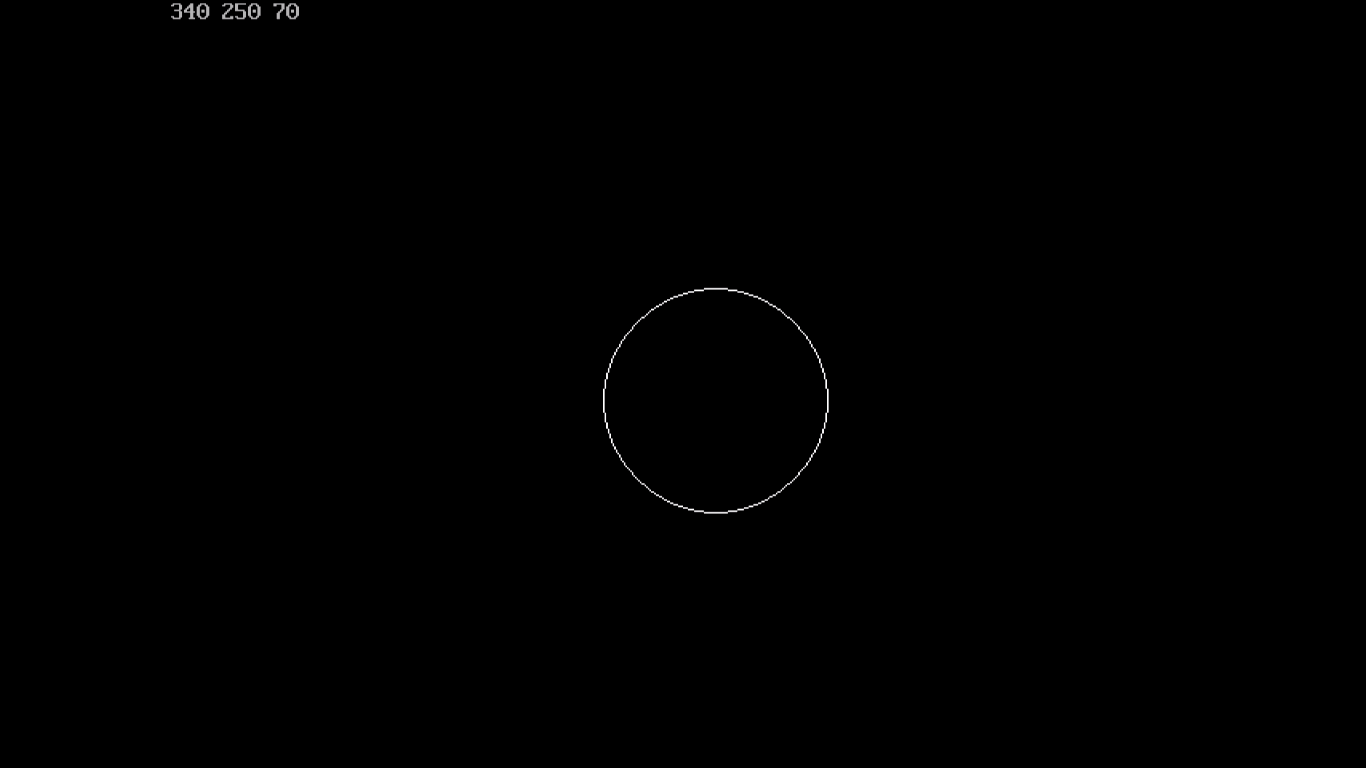
}

getch();

closegraph();

}

**OUTPUT:**

****

**10 Write a program to implement translation in C.**

#include<stdio.h>

#include<graphics.h>

#include<conio.h>

int main(){

int x0,y0,x1,y1,tx,ty;

int gd=DETECT,gm;

initgraph(&gd,&gm,"C://TURBOC3//BGI");

printf("Enter coordinates of first point:\t");

scanf("%d%d",&x0,&y0);

printf("Enter coordinates of second point:\t");

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

printf("Enter coordinates of translation:\t");

scanf("%d%d",&tx,&ty);

setcolor(RED);

line(x0,y0,x1,y1);

setcolor(GREEN);

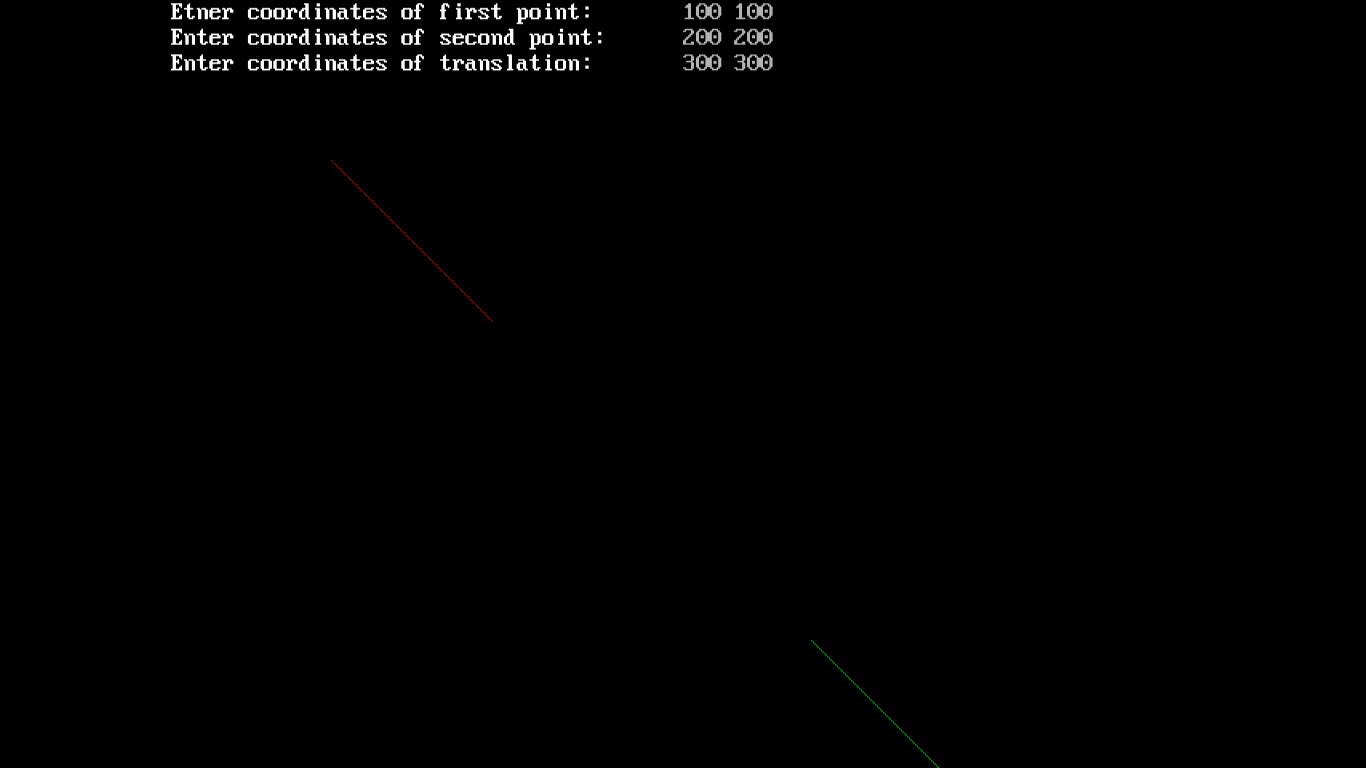
line(x0+tx,y0+ty,x1+tx,y1+ty);

getch();

closegraph();

}

**OUTPUT:**

****

**11 Write a program to implement scale image in C.**

#include<stdio.h>

#include<conio.h>

#include<graphics.h>

int main(){

int a[2],b[2],c[2],scale[2],i;

int gd=DETECT,gm;

initgraph(&gd,&gm,"C://TURBOC3//BGI");

printf("Enter value of a:\t");

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

printf("Enter value of b:\t");

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

printf("Enter value of c:\t");

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

printf("Enter value of scale:\t");

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

setcolor(RED);

line(b[0],b[1],c[0],c[1]);

line(a[0],a[1],b[0],b[1]);

line(a[0],a[1],c[0],c[1]);

setcolor(GREEN);

a[0]=scale[0]\*a[0];

a[1]=scale[1]\*a[1];

b[0]=scale[0]\*b[0];

b[1]=scale[1]\*b[1];

c[0]=scale[0]\*c[0];

c[1]=scale[1]\*c[1];

line(b[0],b[1],c[0],c[1]);

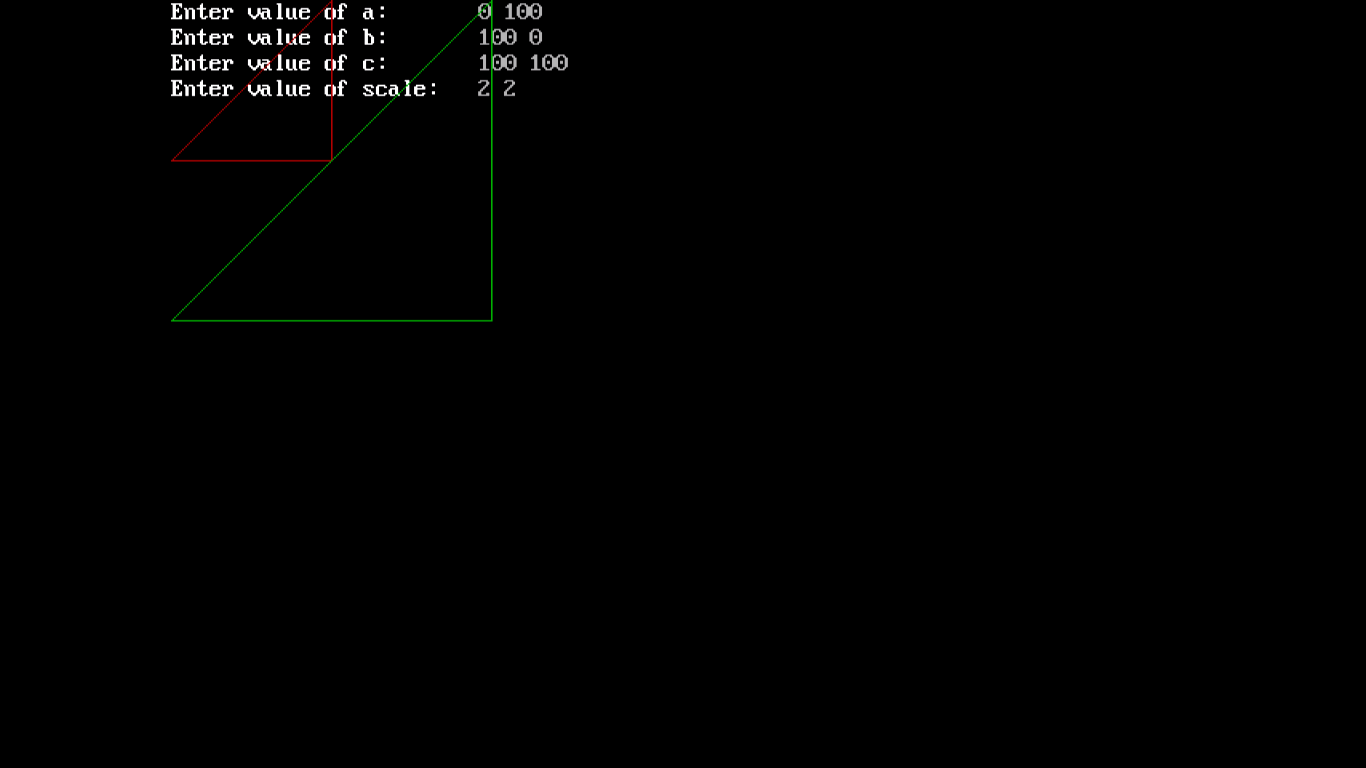
line(a[0],a[1],b[0],b[1]);

line(a[0],a[1],c[0],c[1]);

getch();

}

**OUTPUT:**

****

**12 Write a program to implement Reflection in C.**

#include<stdio.h>

#include<conio.h>

#include<graphics.h>

#define originx 640/2

#define originy 480/2

int main(){

int gd=DETECT,gm;

int x0=20,y0=400,x1=400,y1=300;

initgraph(&gd,&gm,"C://TURBOC3//BGI");

putpixel(originx,originy,RED);

line(x0,y0,x1,y1);

setcolor(GREEN);

line(x0/2+originx,y0/2+originy,x1/2+originx,y1/2+originy);

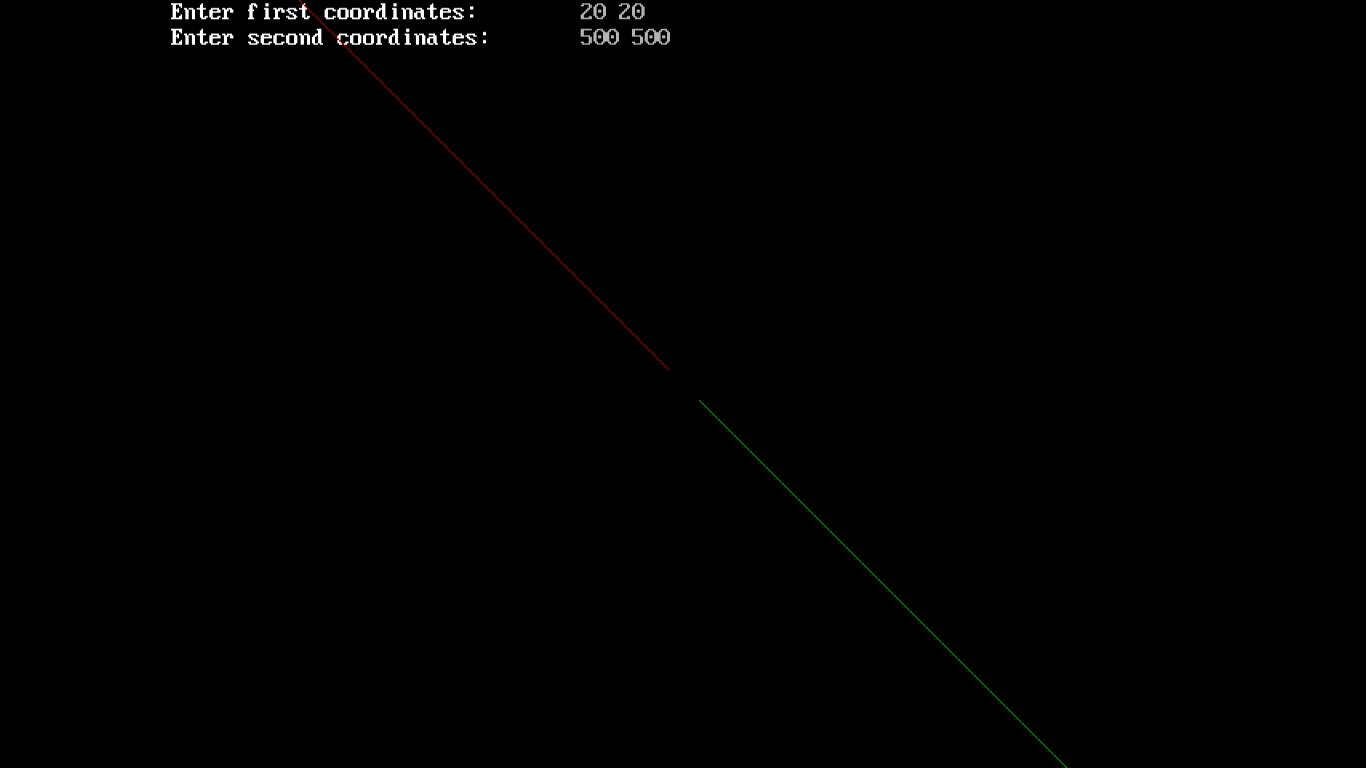
setcolor(RED);

line(-x0/2+originx,-y0/2+originy,-x1/2+originx,-y1/2+originy);

getch();

}

**OUTPUT:**



**13 Write a program to implement shearing in C.**

#include<stdio.h>

#include<graphics.h>

#include<conio.h>

int main(){

int gd=DETECT,gm;

int x0,y0,x1,y1,sx,sy;

initgraph(&gd,&gm,"C://TURBOC3//BGI");

printf("Enter first element:\n");

scanf("%d%d", &x 0, &y0);

printf("\nEnter second element:\n");

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

printf("\nEnter shearing value:\n");

scanf("%d%d", &sx, &sy);

line(x0,y0,x1,y1);

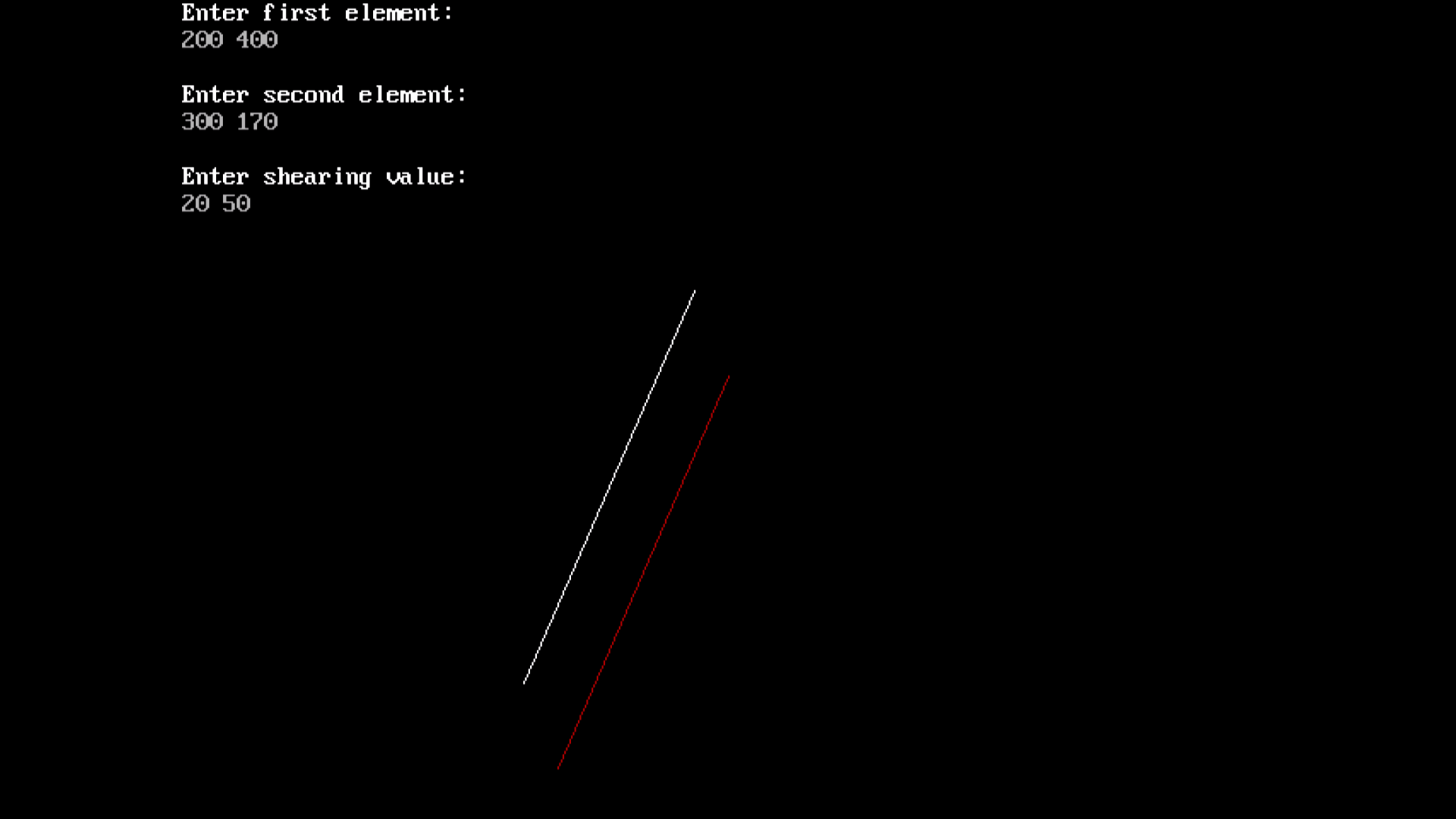
setcolor(RED);

line(x0+sx,y0+sy,x1+sx,y1+sy);

getch();

}

**OUTPUT:**



**14 Write a program to implement rotation in C.**

#include<stdio.h>

#include<graphics.h>

#include<conio.h>

#include<math.h>

int main(){

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

double s,c, angle;

initgraph(&gd,&gm,"C://TURBOC3//BGI");

setcolor(RED);

printf("Enter coordinates of line: ");

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

cleardevice();

setbkcolor(WHITE);

line(x1,y1,x2,y2);

getch();

setbkcolor(BLACK);

printf("Enter rotation angle: ");

scanf("%lf", &angle);

setbkcolor(WHITE);

c = cos(angle \*3.14/180);

s = sin(angle \*3.14/180);

x1 = floor(x1 \* c - y1 \* s);

y1 = floor(x1 \* s + y1 \* c);

x2 = floor(x2 \* c -

y2 \* s);

y2 = floor(x2 \* s + y2 \* c);

cleardevice();

line(x1, y1 ,x2, y2);

printf("%d %d %d %d",x1,y1,x2,y2);

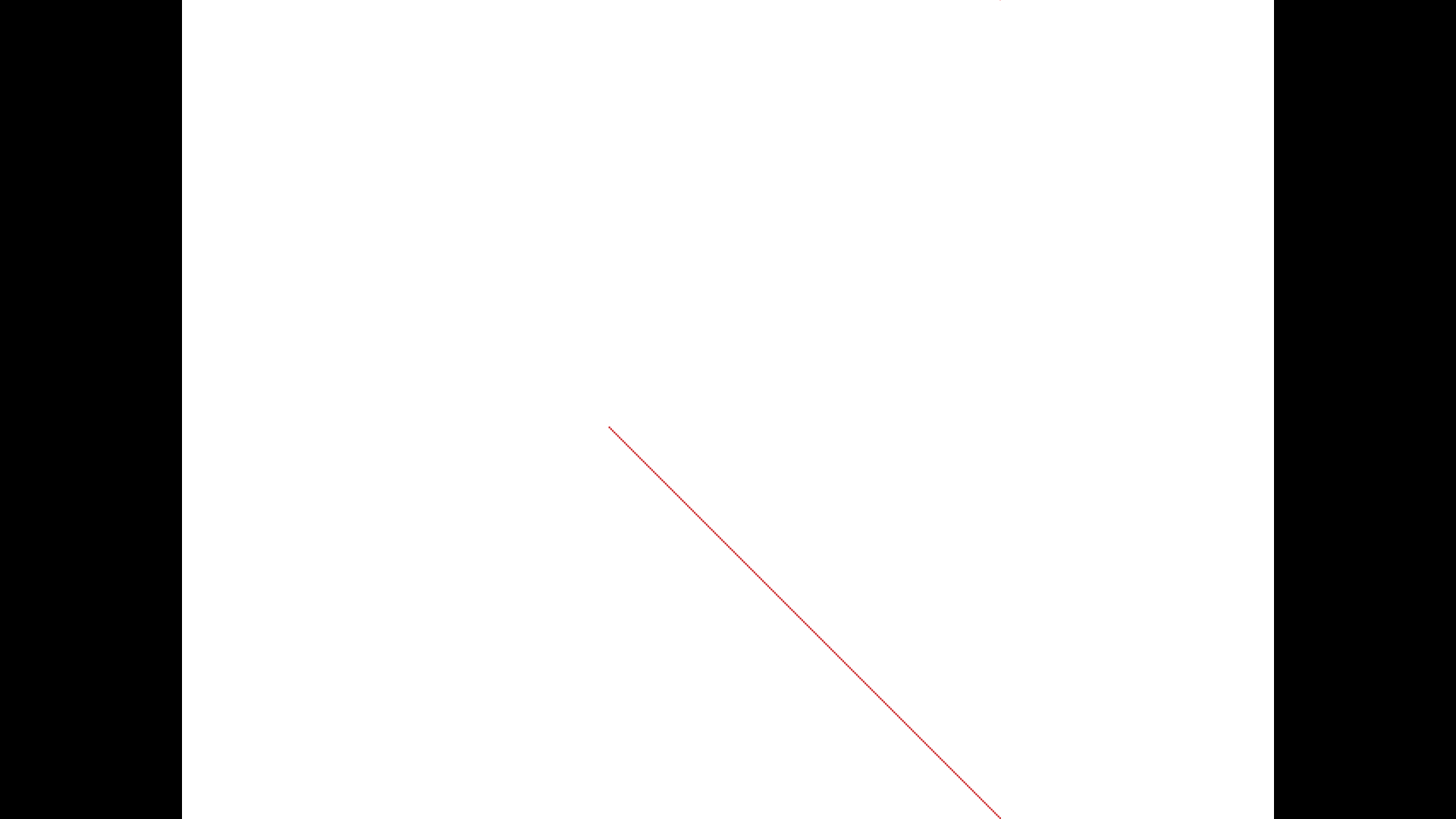
getch();

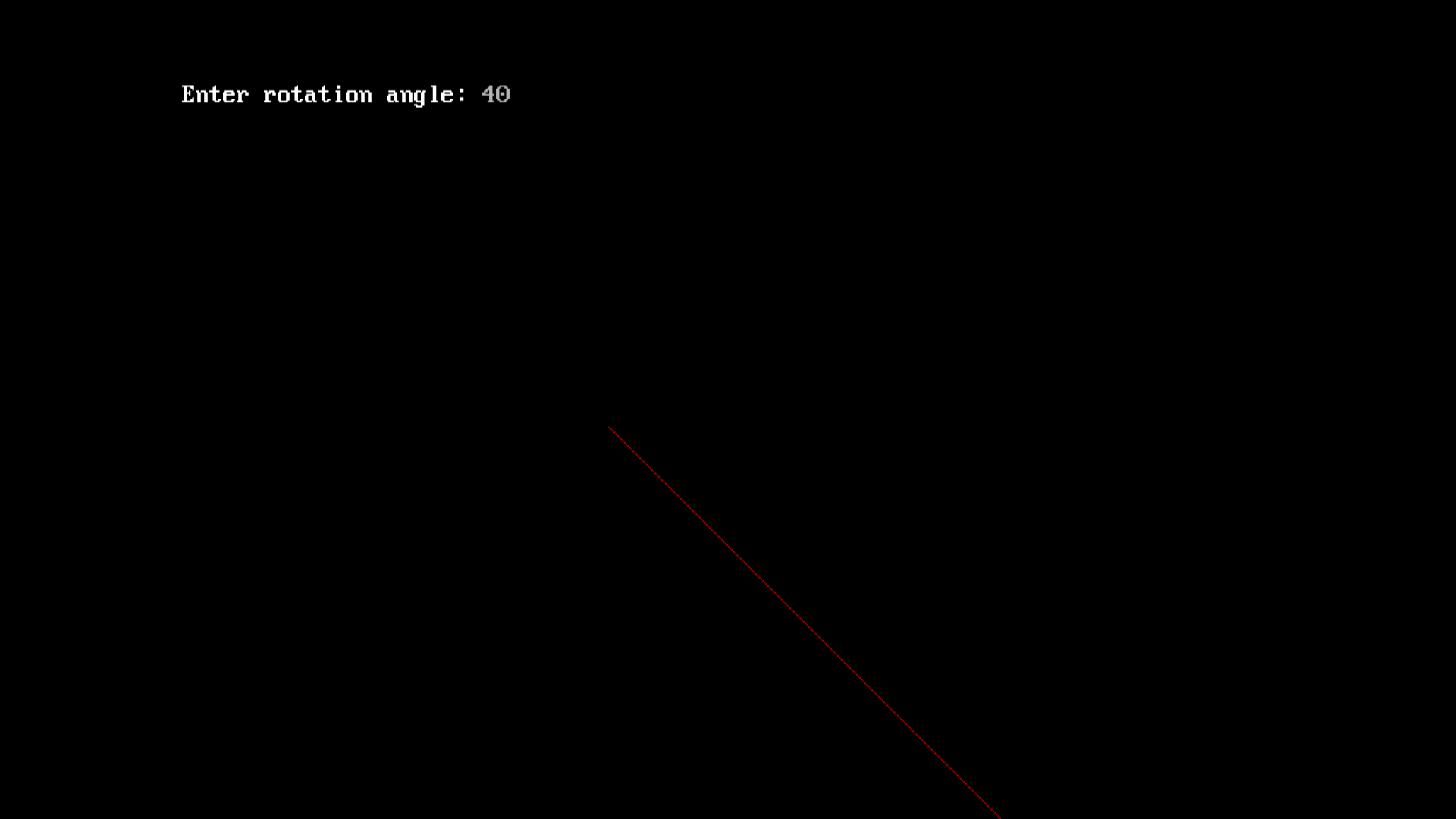
closegraph();

}

**OUTPUT**









**15. Write a program in C to implement floodfill algorithm.**

#include<stdio.h>

#include<graphics.h>

#include<conio.h>

void floodFill(int x,int y,int newCol,int oldCol){

int pixColor=getpixel(x,y);

if(pixColor==oldCol){

putpixel(x,y,newCol);

floodFill(x+1,y,newCol,oldCol);

floodFill(x-1,y,newCol,oldCol);

floodFill(x,y+1,newCol,oldCol);

floodFill(x,y-1,newCol,oldCol);

}

}

int main(){

int gd=DETECT,gm;

initgraph(&gd,&gm,"C://TURBOC3//BGI");

rectangle(100,100,150,190);

floodFill(149,149,RED,0);

getch();

return 0;

}

**OUTPUT:**

