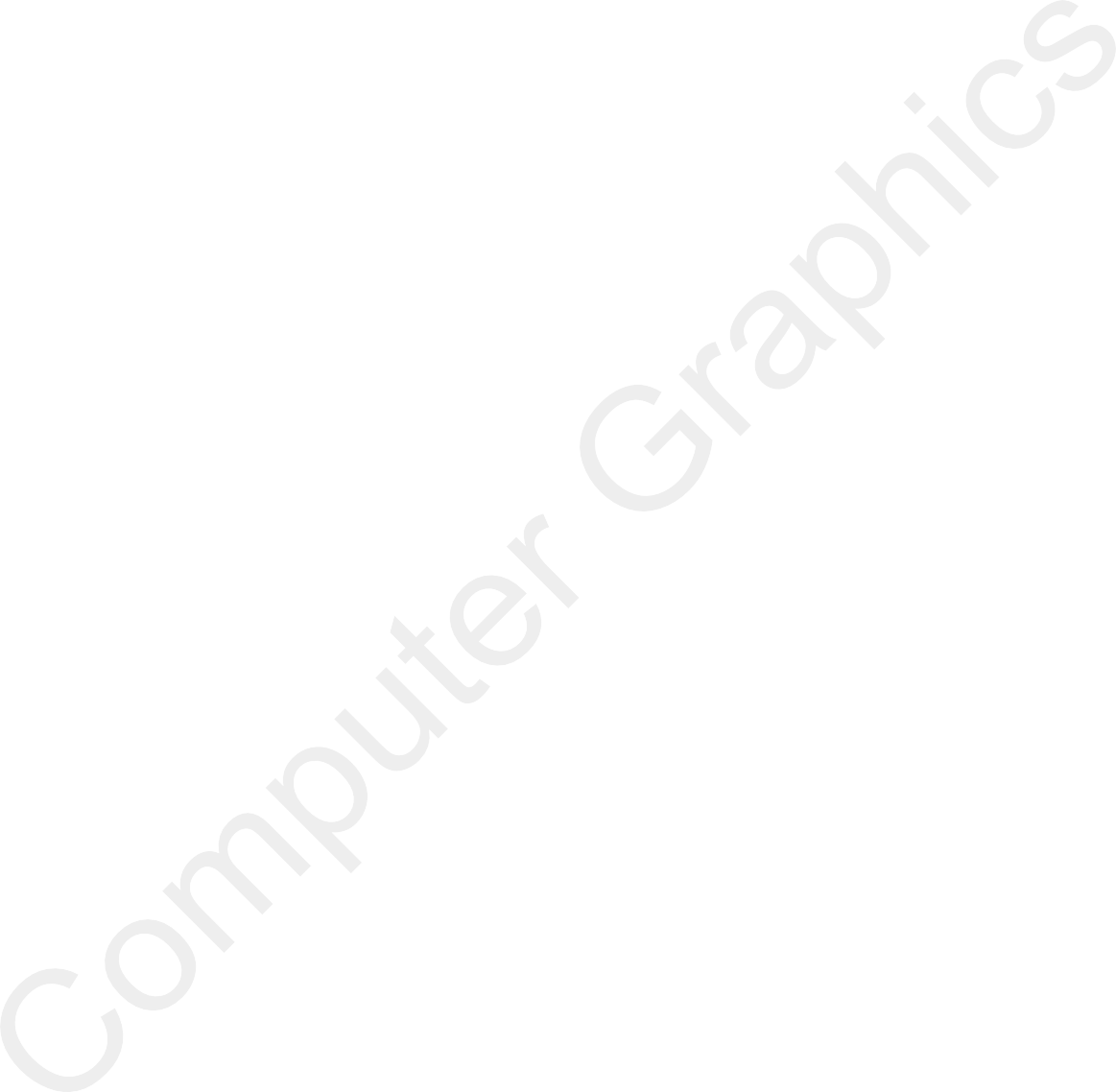
**INDEX**

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Program Name** | **Remarks** |
| **1.** | Write a Program for Simple DDA. |  |
| **2.** | Write a Program for Symitrical DDA. |  |
| **3.** | Write a Program for Bresenham’s Line  Drawing |  |
| **4.** | Write a Program for Bresenham’s Circle  Drawing. |  |
| **5.** | Write a Program for Midpoint Circle. |  |
| **6.** | Write a Program for Rotation(Tringle, Line,  Rectangle) |  |
| **7.** | Write a Program for Translation(Tringle,  Line, Rectangle) |  |
| **8.** | Write a Program for Shearing(Tringle, Line,  Rectangle) |  |
| **9.** | Write a Program for cohen sutherland line  clipping. |  |
| **10.** | Write a Program for Midpoint line clipping  Subdivision. |  |

**Sign………………**

# Que(1):Write a Program for Simple DDA. Ans:-

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

#define round(a)((int)(a+0.5)) void main()

{ //variable type deration

int i,dx,dy,steps,p,q,x,y,xinc,yinc,x1,x2,y1,y2,gd=DETECT,gm; x1=100,y1=200;x2=500;y2=505;

initgraph(&gd,&gm,"c:\\tc\\bgi"); //point defined dx=x2-x1;

dy=y2-y1; if(abs(dx)>abs(dy)) steps=abs(dx);

else steps=abs(dy);

xinc=(float)dx/steps; yinc=(float)dy/steps; x=x1;

y=y1; putpixel(x,y,7); for(i=1;i<steps;i++)

{

x=x+xinc; //x=x+(1/m) y=y+yinc;

p=round(x);

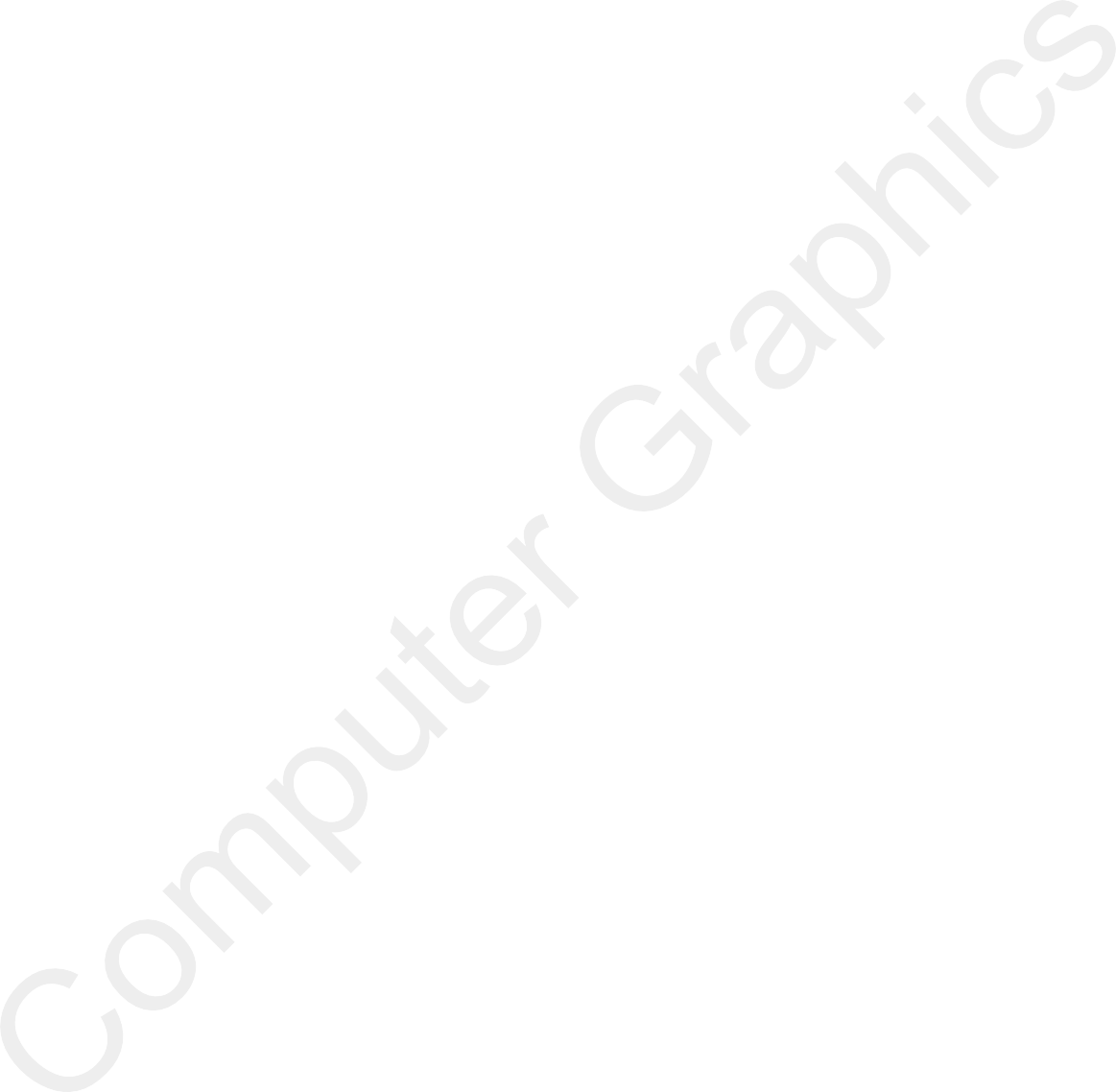
q=round(y); //y=y+m ; putpixel(p,q,7);

}

getch(); closegraph();

}

# Que(2):Write a Program for Symitrical DDA. Ans:-

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

{

int x1,y1,x2,y2,i,len,gdriver=DETECT,gmode; float incx,incy,x,y;

clrscr(); initgraph(&gdriver,&gmode,"c:\\TC\\BGI"); cout<<"enter the value of x1,y1,x2,y2"<<endl; cin>>x1>>y1>>x2>>y2;

len=abs(x2-x1); if(abs(y2-y1)>len) len=abs(y2-y1); incx=(x2-x1)/len; incy=(y2-y1)/len; x=x1+0.5; y=y1+0.5;

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

{

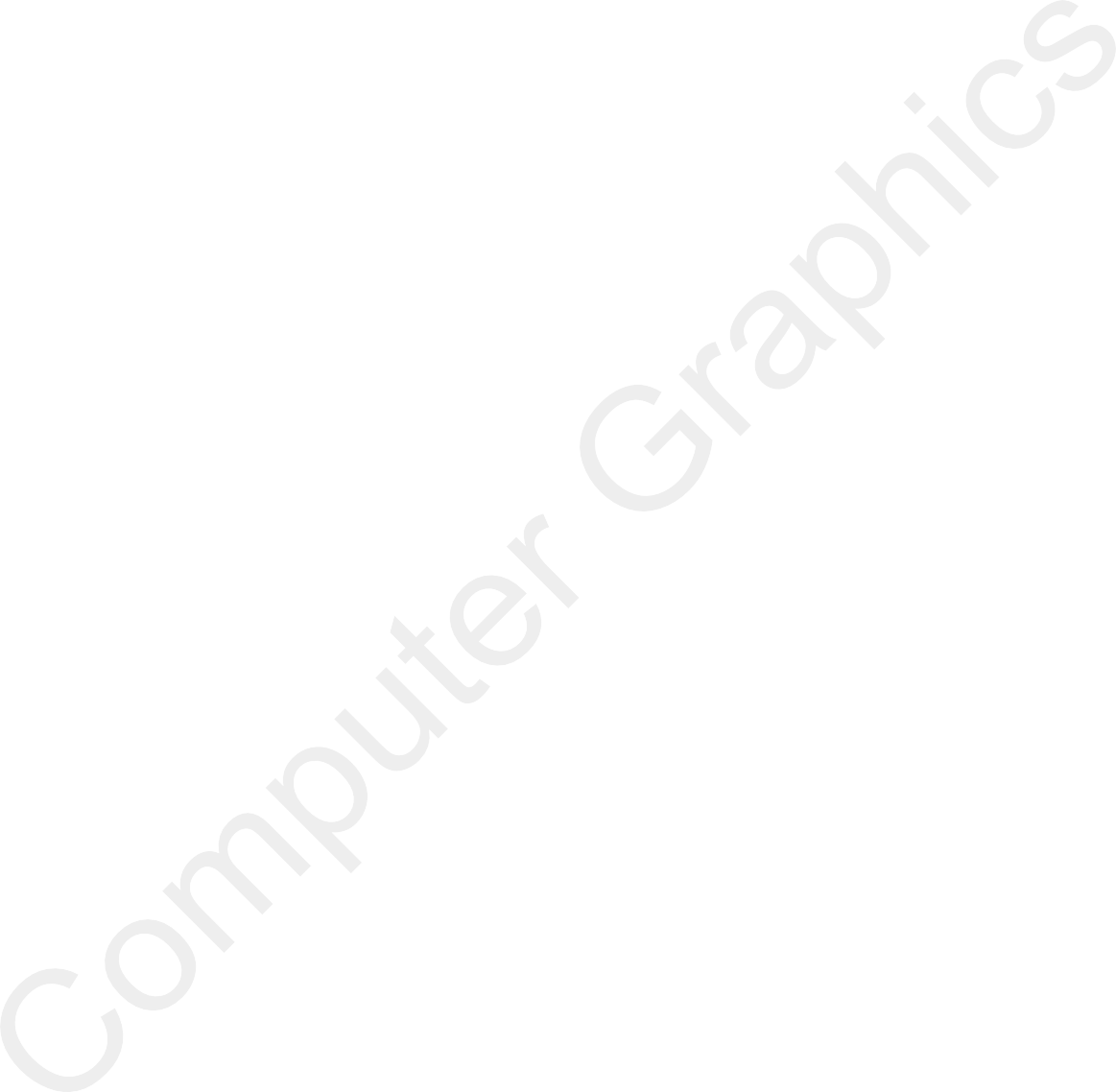
putpixel(x,y,9); x=x+incx; y=y+incy; delay(100);

}

getch();

}

# Que(3):Write a Program for Bresenham’s Line Drawing Ans:-

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

{

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

int i,flag,d; clrscr();

cout<<"Enter value of(x1,y1)="; cin>>x1>>y2;

cout<<"Enter value of(x2,y2)="; cin>>x2>>y2; initgraph(&gd,&gm,"c:\\TC\\BGI"); int dx,dy;

dx=abs(x2-x1); dy=abs(y2-y1); int x,y,t,s1,s2; x=x1;

y=y1;

if((x2-x1)>0) s1=1;

else s1=-1;

if(dy-dx)

{

t=dx; dx=dy; dy=t;

}

else

flag=0; d=2\*dy-dx;

outtextxy(x1,y1,"(x1,y1)");

outtextxy(x2,y2,"(x2,y2)"); i=1;

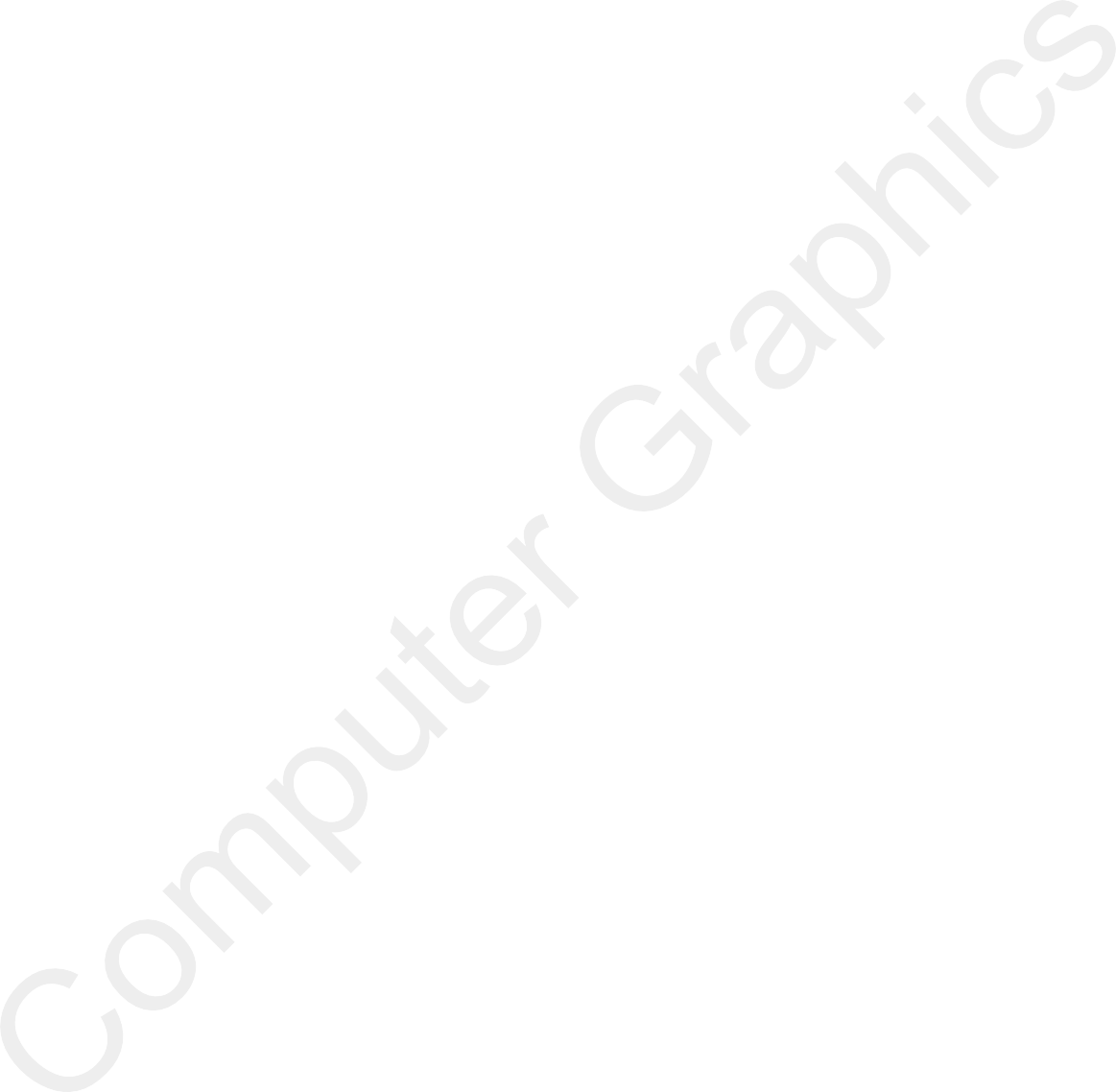
a: putpixel(x,y,3);

delay(40); while(d>=0)

{

getch();

if(flag==1) x=x+s1; d=d+2\*dy; i++;

if(i<=dx) goto a;

}

closegraph();

}

# Que(4):Write a Program for Bresenham’s Circle Drawing. Ans:-

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

{

int x1,y1,x,y,r,p,gdrive=DETECT,gmode; clrscr(); initgraph(&gdrive,&gmode,"c:\\TC\\BGI"); cout<<"Enter the value of x1,y1,r"<<endl; cin>>x1>>y1>>r;

x=0;

y=r;

p=3-2\*r; while(x<=y)

{

putpixel(x1+x,y1+y,2); putpixel(x1-x,y1+y,2); putpixel(x1+x,y1-y,2); putpixel(x1-x,y1-y,2); putpixel(x1+y,y1+x,2); putpixel(x1-y,y1+x,2); putpixel(x1+y,y1-x,2); putpixel(x1-y,y1-x,2); if(p<0) p=p+(4\*x++)+6;

else

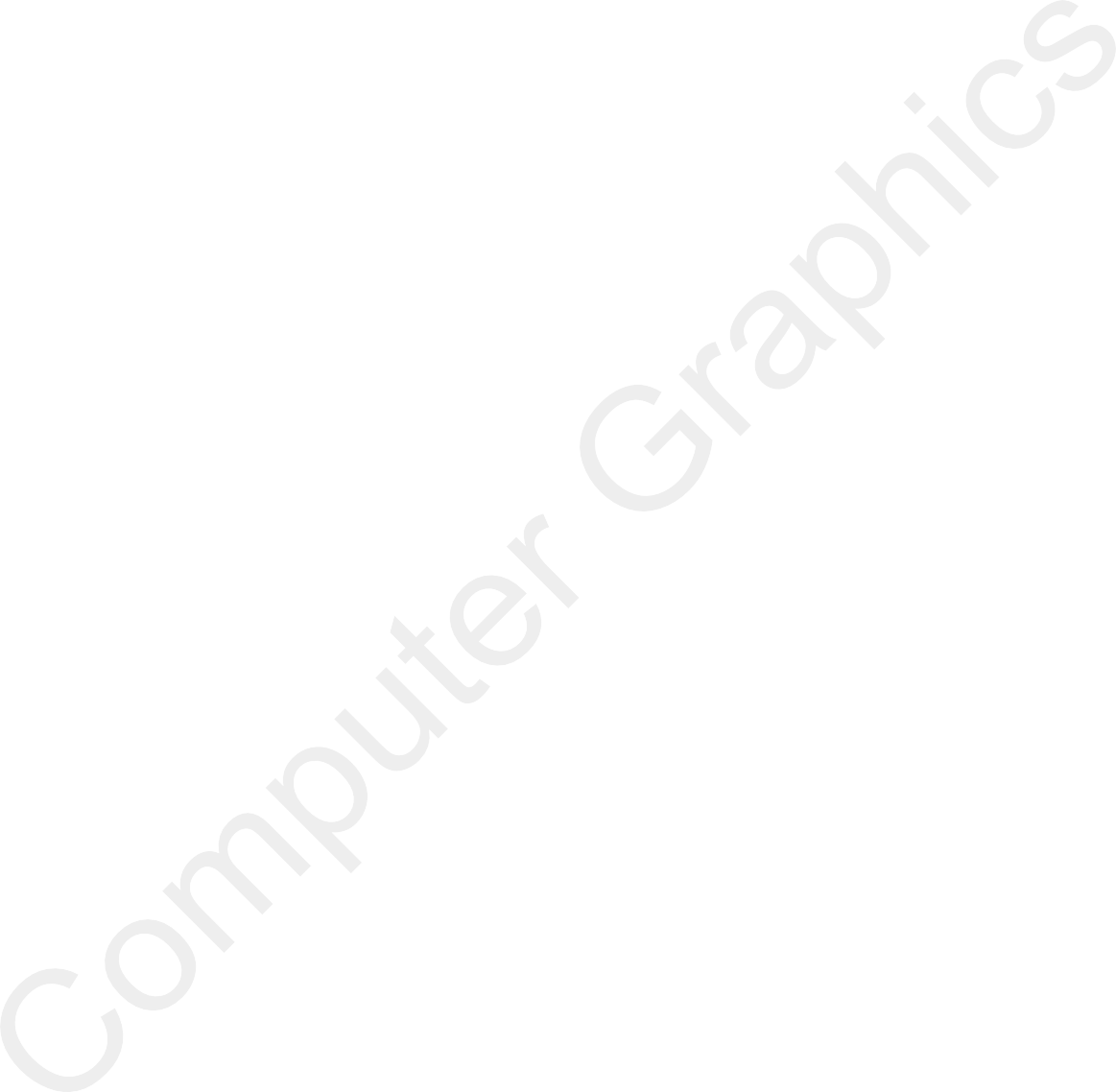
p=p+4\*(x++-y--)+10;

}

getch(); closegraph;

}

# Que(5):Write a Program for Midpoint Circle. Ans:-



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

void pixel(int xc,int yc,int x,int y); void main()

{

int gd=DETECT,gm,xc,yc,r,x,y,Pk; clrscr(); initgraph(&gd,&gm,"c:\\turboc3\\bgi ");

printf("\*\*\* Mid-Point Subdivision algorithm of circle \*\*\*\n"); printf("Enter the value of Xc\t");

scanf("%d",&xc);

printf("Enter the value of Yc \t"); scanf("%d",&yc);

printf("Enter the Radius of circle\t"); scanf("%d",&r);

x=0;

y=r; Pk=1-r;

pixel(xc,yc,x,y); while(x<y)

{

if(Pk<0)

{

}

else

{

}

x=x+1; Pk=Pk+(2\*x)+1;

x=x+1; y=y-1;

Pk=Pk+(2\*x)-(2\*y)+1;

pixel(xc,yc,x,y);

}

getch(); closegraph();

}

void pixel(int xc,int yc,int x,int y)

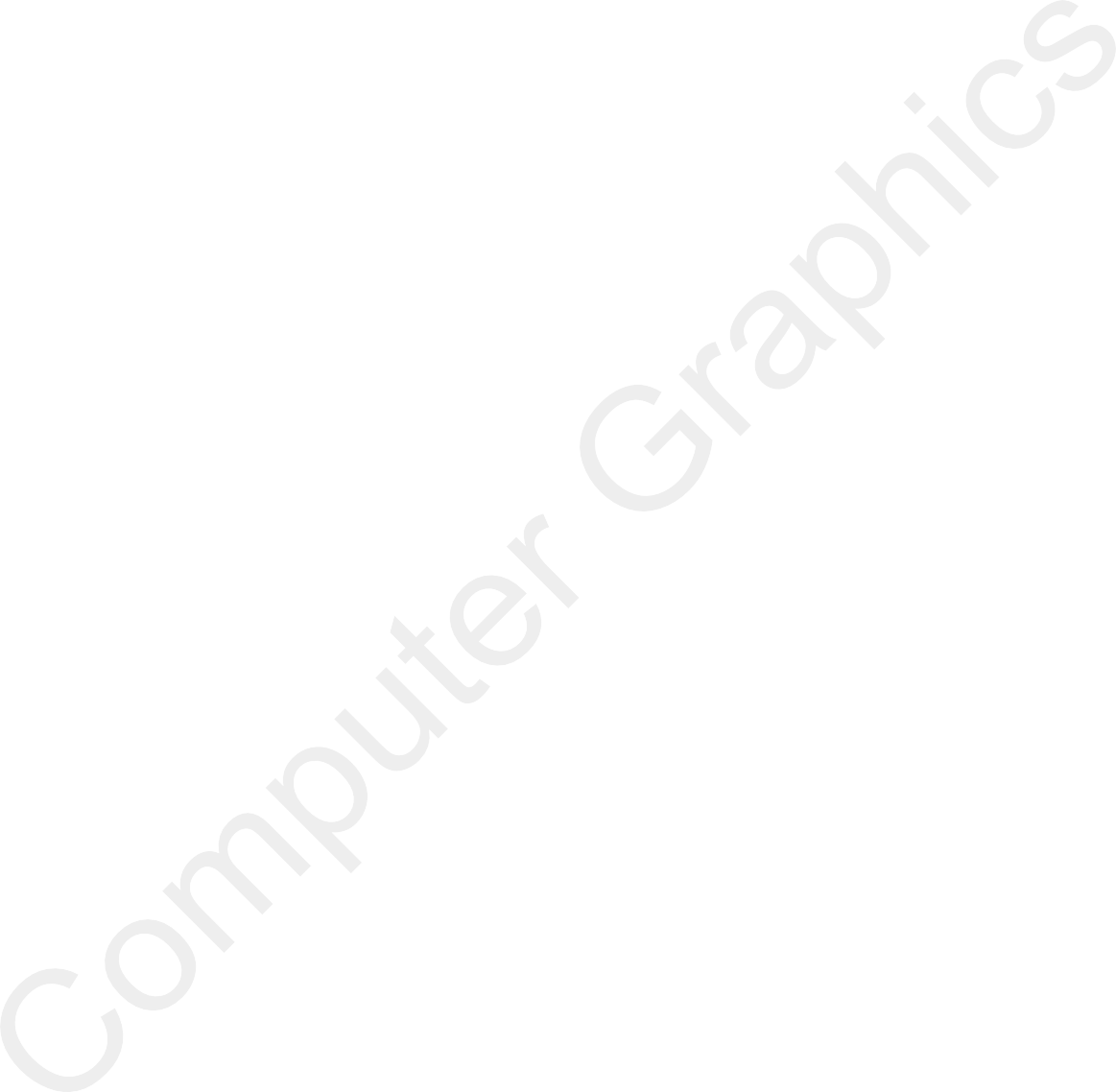
{

putpixel(xc+x,yc+y,7); putpixel(xc+y,yc+x,7); putpixel(xc-y,yc+x,7); putpixel(xc-x,yc+y,7); putpixel(xc-x,yc-y,7);

putpixel(xc-y,yc-x,7); putpixel(xc+y,yc-x,7); putpixel(xc+x,yc-y,7);

}

# Que(6):Write a Program for Rotation (Tringle, Line, Rectangle) Ans:-



**Rotation of triangle in graphics** #include<stdio.h> #include<graphics.h>

#include<math.h>

main()

{

intgd=0,gm,x1,y1,x2,y2,x3,y3; double s,c, angle;

initgraph(&gd, &gm, "C:\\TURBOC3\\BGI"); setcolor(RED);

printf("Enter coordinates of triangle: "); scanf("%d%d%d%d%d%d",&x1,&y1,&x2,&y2, &x3, &y3); setbkcolor(WHITE);

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

line(x2,y2, x3,y3);

line(x3, y3, x1, y1); getch(); setbkcolor(BLACK);

printf("Enter rotation angle: "); scanf("%lf", &angle); setbkcolor(WHITE);

c = cos(angle \*M\_PI/180); s = sin(angle \*M\_PI/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); x3 = floor(x3 \* c + y3 \* s); y3 = floor(-x3 \* s + y3 \* c);

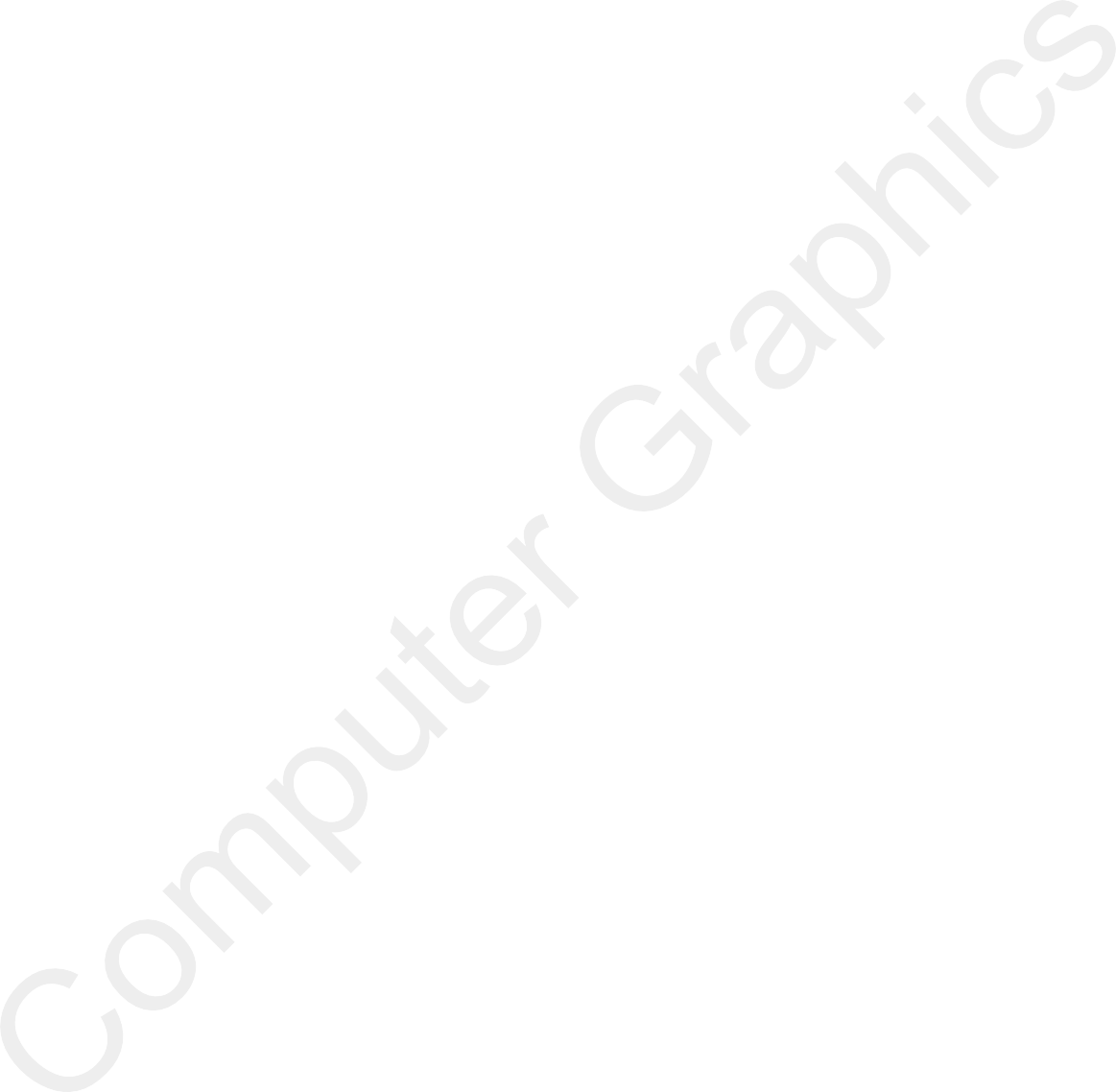
cleardevice();

line(x1, y1 ,x2, y2);

line(x2,y2, x3,y3);

line(x3, y3, x1, y1); getch();

closegraph(); return 0;

}

**Line rotarion program** #include<stdio.h> #include<graphics.h>

void main()

{

int gd=DETECT,gm; int x1,y1,x2,y2 ; float b1,b2;

float t,deg; initgraph(&gd,&gm,”c:\\tc\\”); printf(“Enter the coordinates of Line \n”);

scanf(“%d%d%d%d”,&x1,&y1,&x2,&y2); setcolor(6);

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

printf(“Enter the angle of rotation: “); scanf(“%f”,&deg); t=(22\*deg)/(180\*7); b1=abs((x2\*cos(t))-(y2\*sin(t)));

b2=abs((x2\*sin(t))+(y2\*cos(t))); line(x1,y1,b1,b2);

getch(); closegraph();

}

# Que(7):Write a Program for Translation (Tringle, Line, Rectangle) Ans:-

**Translation (Tringle)** #include<conio.h> #include<graphics.h> #include<stdio.h>

void main()

{

int gd=DETECT,gm;

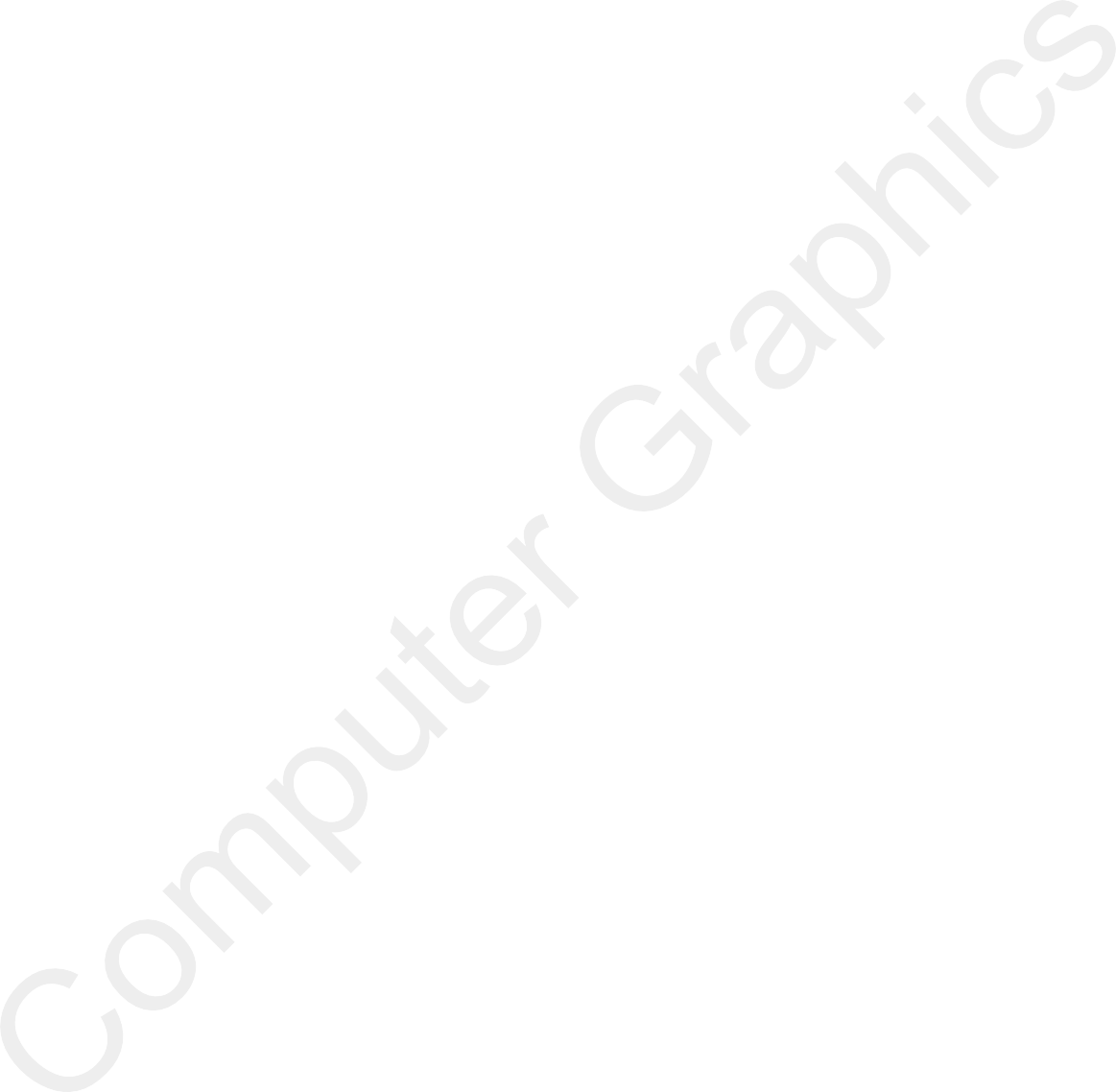
int x,y,x1,y1,x2,y2,tx,ty;

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

printf("\n Please enter first coordinate of the triangle= "); scanf("%d %d", &x,&y);

printf("\n Enter second coordinate of the trinagle = "); scanf("%d %d",&x1,&y1);

printf("\n Enter third coordinate of the triangle = "); scanf("%d %d",&x2,&y2);

printf("\n\t\t\*\*\*\*\*\*\*\*\*\* TRIANGLE before & after translation \*\*\*\*\*\*\*\*\*\*\*"); line(x,y,x1,y1);

line(x1,y1,x2,y2);

line(x2,y2,x,y);

printf("\n Now enter the translation vector = "); scanf("%d %d",&tx,&ty);

setcolor(RED); line(x+tx,y+ty,x1+tx,y1+ty); line(x1+tx,y1+ty,x2+tx,y2+ty); line(x2+tx,y2+ty,x+tx,y+ty); getch();

closegraph();

}

**Translation (Line)** #include<stdio.h> #include <graphics.h> #include <stdlib.h>

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

{

int gdriver = DETECT, gmode, errorcode; int xmax, ymax,x1,y1,x2,y2,tx,ty; initgraph(&gdriver, &gmode, “”);

printf(“Enter the X1 coordinate:\n”); scanf(“%d”,&x1);

printf(“Enter the Y1 coordinate:\n”); scanf(“%d”,&y1);

printf(“Enter the X2 coordinate:\n”); scanf(“%d”,&x2);

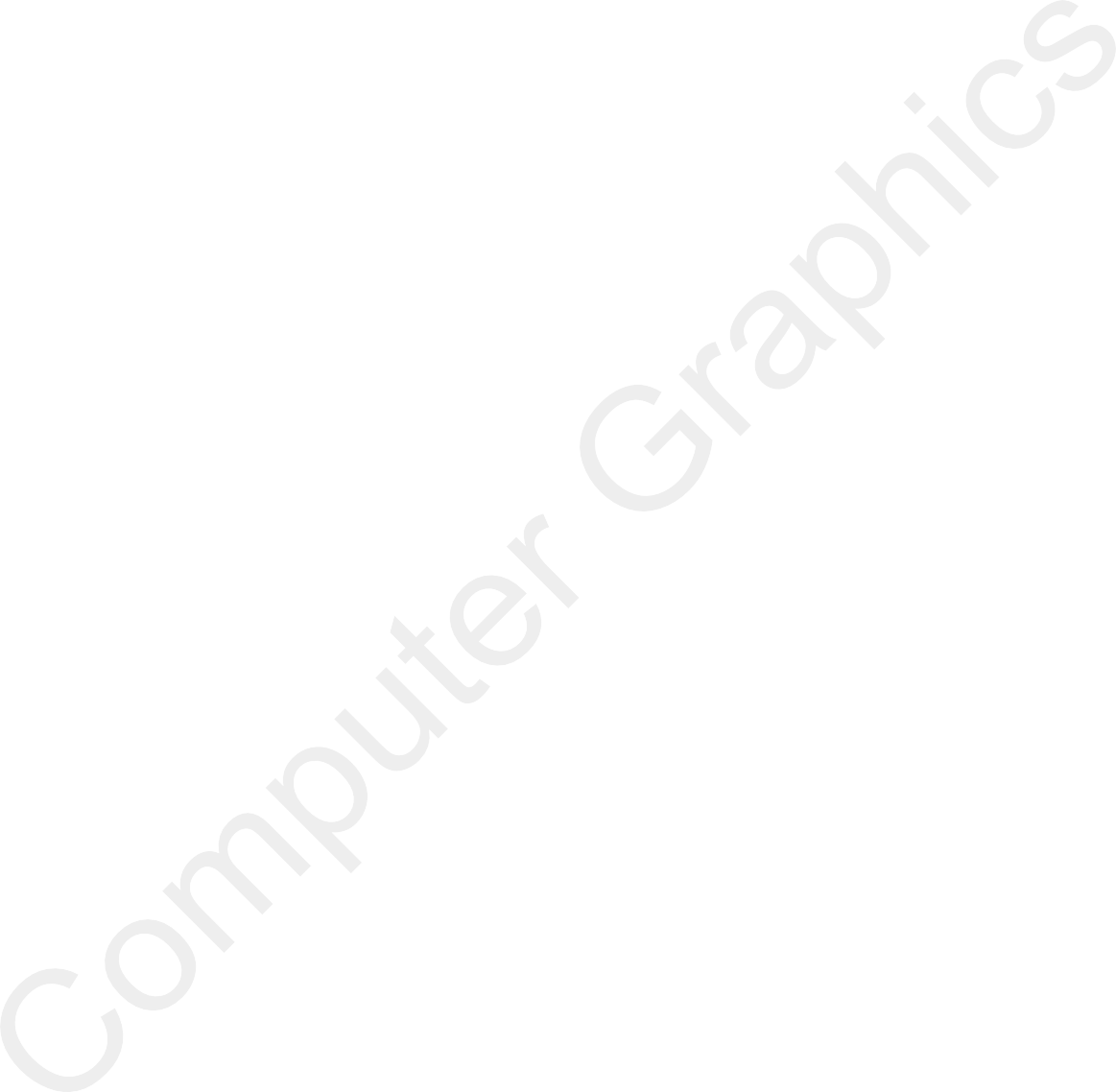
printf(“Enter the Y2 coordinate:\n”); scanf(“%d”,&y2);

line(x1,y1,x2,y2);

printf(“Enter the translation vector:\n”); printf(“tx:”);

scanf(“%d”,&tx);

printf(“ty:”);



scanf(“%d”,&ty); line(x1+tx,y1+ty,x2+tx,y2+ty); getch();

closegraph(); return 0;

}

# Translation (Rectangle)

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

void RectAngle(int x, int y, int Height, int Width);

void Translate(int x, int y, int Height, int Width);

void main() {

int gd = DETECT, gm; int x, y, Height, Width; initgraph(&gd, &gm, " ");

printf("Enter the First point for the Rectangle:"); scanf("%d%d", &x, &y);

printf("Enter the Height&Width for the Rectangle:"); scanf("%d%d", &Height, &Width);

RectAngle(x, y, Height, Width); getch();

cleardevice();

Translate(x, y, Height, Width);

RectAngle(x, y, Height, Width); getch();

}

void RectAngle(int x, int y, int Height, int Width) { line(x, y, x + Width, y);

line(x, y, x, y + Height);

line(x + Width, y, x + Width, y + Height); line(x, y + Height, x + Width, y + Height);

}

void Translate(int x, int y, int Height, int Width) { int Newx, Newy, a, b;

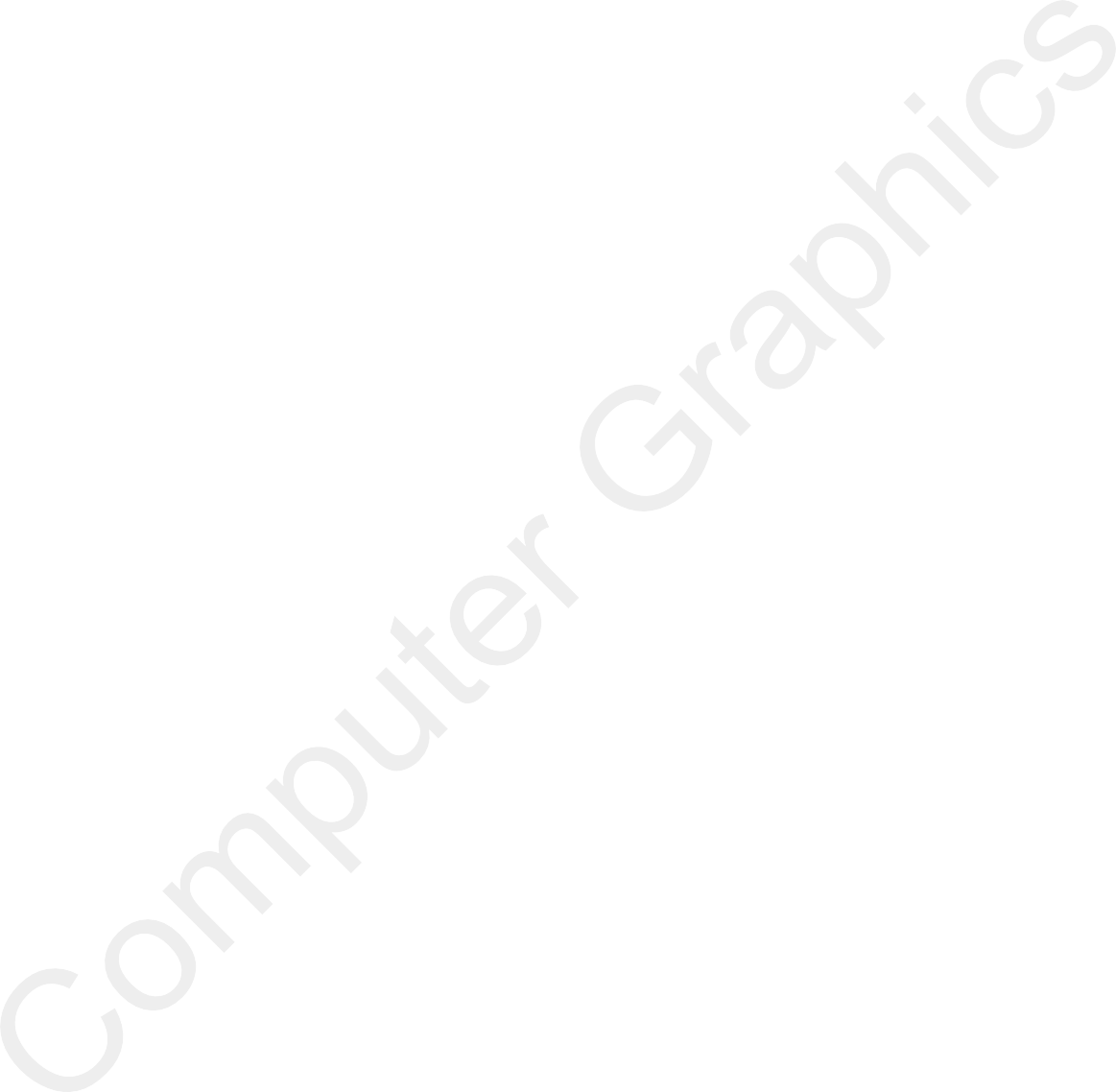
printf("Enter the Transaction coordinates"); scanf("%d%d", &Newx, &Newy); cleardevice();

a = x + Newx; b = y + Newy;

RectAngle(a, b, Height, Width);

}

**Que(8):Write a Program for Shearing (Tringle, Line, Rectangle) Ans:-**



//\*\*\*\*\*\*\*PROGRAM FOR SHEARING\*\*\*\*\*\*//

#include<graphics.h> #include<stdlib.h> #include<stdio.h> #include<conio.h> #include<iostream.h> #include<dos.h> #include<math.h>

int x1,y1,x2,y2,x,y,x3,y3,x4,y4,ch; void main()

{

int gd = DETECT,gm,errorcode; initgraph(&gd,&gm,"c:\\tc\\bgi");

errorcode = graphresult(); if(errorcode!=grOk)

{

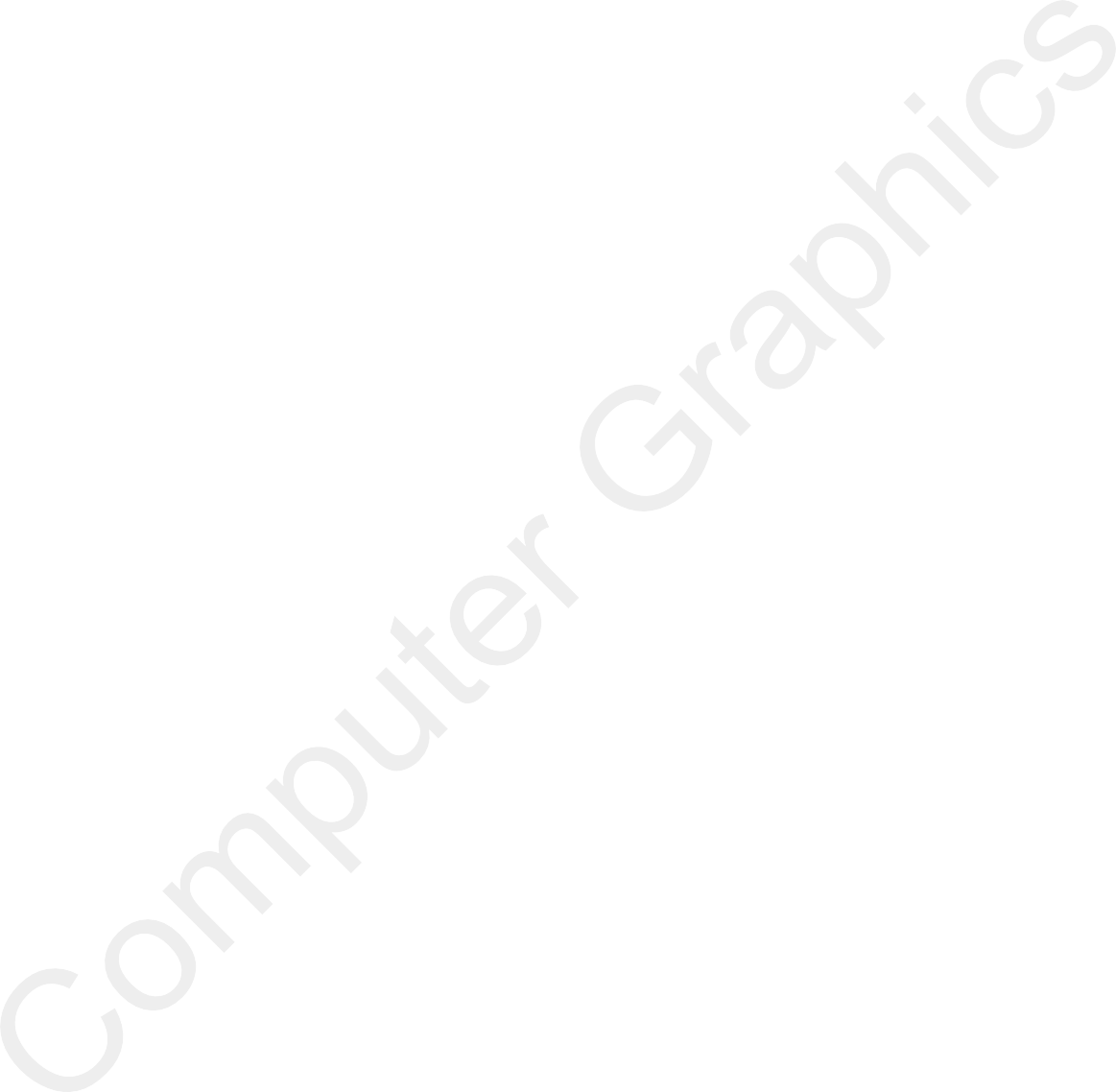
printf("Graphics error:%s\n",grapherrormsg(errorcode)); printf("press any key to halt:");

getch();

exit(1);

}

do

{

clrscr();

cout<<" #############MAIN-ENU###############\n";

cout<<" SHEARING\n"; cout<<" 1.LINE\n";

cout<<" 2.RECTANGLE\n"; cout<<" 3.TRINGLE\n";

cout<<"enter your choice:0 for exit:\n"; cin>>ch;

switch(ch)

{

case 1:

cout<<"enter the value of line coordinates:"; cin>>x1>>y1>>x2>>y2;

cout<<"enter the value of shearing for x-axis:"; cin>>x;

cout<<"enter the value of shearing for y-axis:"; cin>>y;

cleardevice(); setcolor(4); line(x1,y1,x2,y2);

cout<<"now hit a key to see shear in x\_axis:"; getch();

setcolor(1); line(x1,y1,x2\*x,y2);

cout<<"\nnow hit a key to see shear in y\_axis:"; getch();

setcolor(58); line(x1,y1,x2,y2\*y); getch();

break; case 2:

cout<<"enter the top left coordinates:"; cin>>x1>>y1;

cout<<"enter the bottom right coordinates:"; cin>>x2>>y2;

cout<<"enter the value of shearing coordinates for x-shear:"; cin>>x;

cout<<"enter the value of shearing coordinates for y-shear:"; cin>>y;

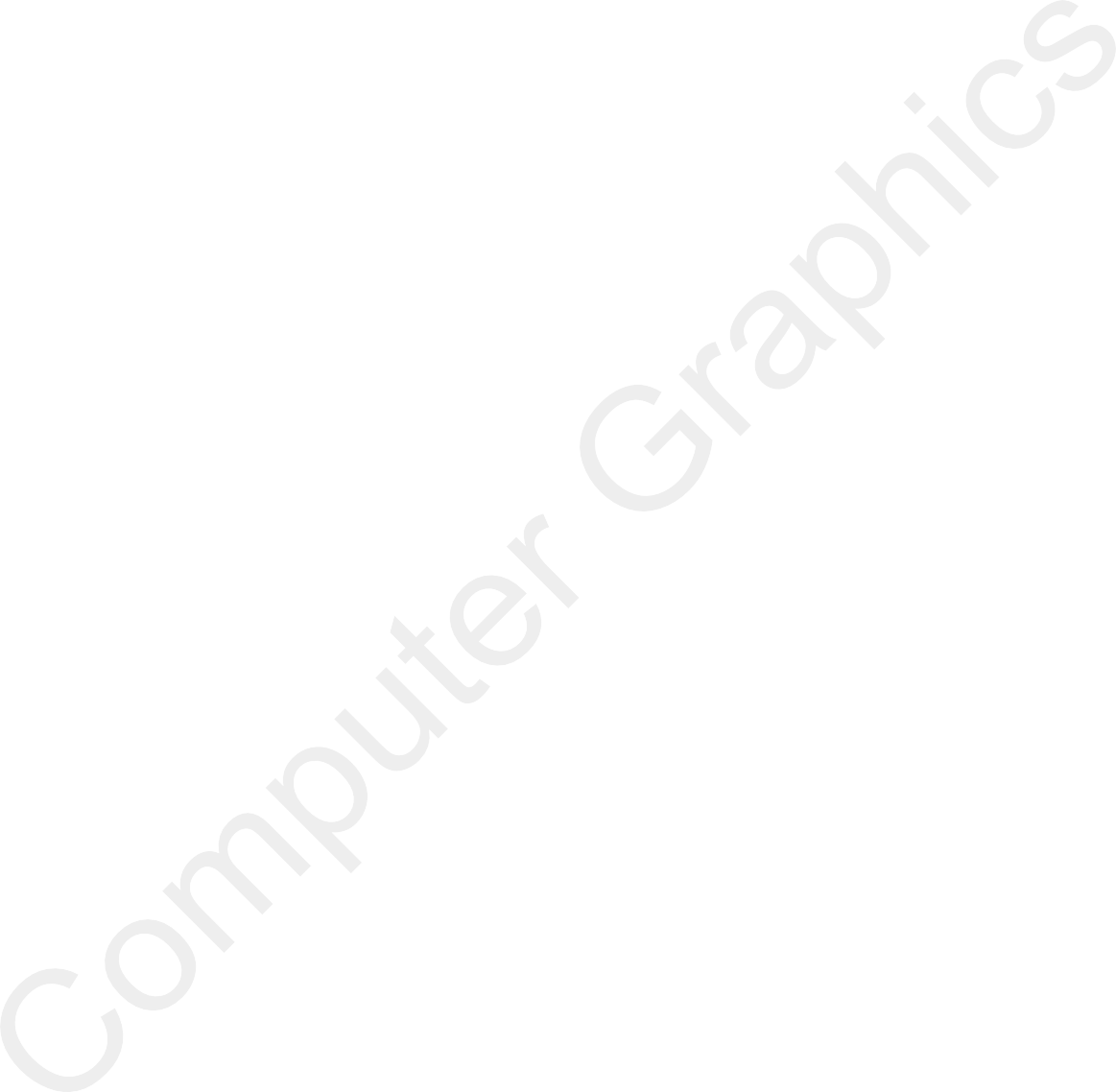
cleardevice();

setcolor(5); rectangle(x1,y1,x2,y2);

cout<<"now hit a key to see shear in x\_axis:"; getch();

setcolor(1); rectangle(x1,y1,x2+x\*y2,y2);

cout<<"\nnow hit a key to see shear in y\_axis:"; getch();

setcolor(58); rectangle(x1,y1,x2,y2+y\*x2); getch();

break; case 3:

cout<<"enter the coordinates of triangle:\n"; cin>>x1>>y1>>x2>>y2;

cin>>x3>>y3;

cout<<"enter the value of shearing coordinates for x-shear:"; cin>>x;

cout<<"enter the value of shearing coordinates for y-shear:"; cin>>y;

cleardevice(); setcolor(5); line(x1,y1,x2,y2);

line(x1,y1,x3,y3);

line(x3,y3,x2,y2);

cout<<"\nnow hit a key to see shear in x\_axis:"; getch();

setcolor(10); line(x1,y1,x2\*x,y2); line(x1,y1,x3\*x,y3); line(x3\*x,y3,x2\*x,y2);

cout<<"\nnow hit a key to see shear in y\_axis:"; getch();

setcolor(1); line(x1,y1,x2,y2\*y); line(x1,y1,x3,y3\*y); line(x3,y3\*y,x2,y2\*y); getch();

break; case0:break;

default:cout<<"invalid choice";break;

}}while(ch!=0); closegraph(); getch();

}

# Que(9):Write a Program for cohen sutherland line clipping. Ans:-

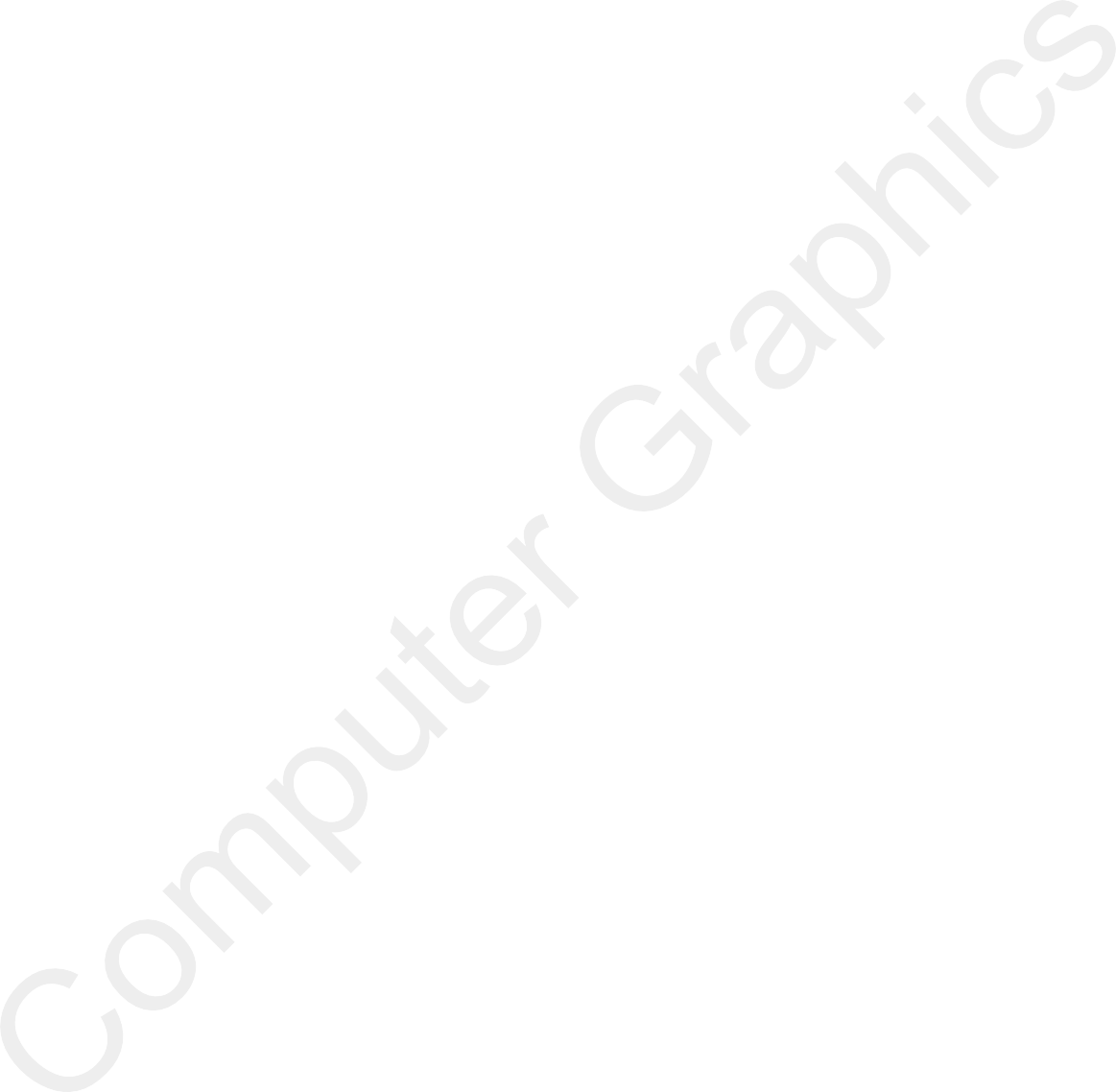
#include<stdio.h> #include<stdlib.h> #include<math.h> #include<graphics.h> #include<dos.h>

typedef struct coordinate

{

}PT;

int x,y;

char code[4];

void drawwindow();

void drawline(PT p1,PT p2); PT setcode(PT p);

int visibility(PT p1,PT p2); PT resetendpt(PT p1,PT p2);

void main()

{

int gd=DETECT,v,gm; PT p1,p2,p3,p4,ptemp;

printf("\nEnter x1 and y1\n"); scanf("%d %d",&p1.x,&p1.y); printf("\nEnter x2 and y2\n"); scanf("%d %d",&p2.x,&p2.y);

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

delay(500);

drawline(p1,p2); delay(500); cleardevice();

delay(500); p1=setcode(p1); p2=setcode(p2); v=visibility(p1,p2); delay(500);

switch(v)

{

case 0: drawwindow();

delay(500); drawline(p1,p2); break;

case 1: drawwindow();

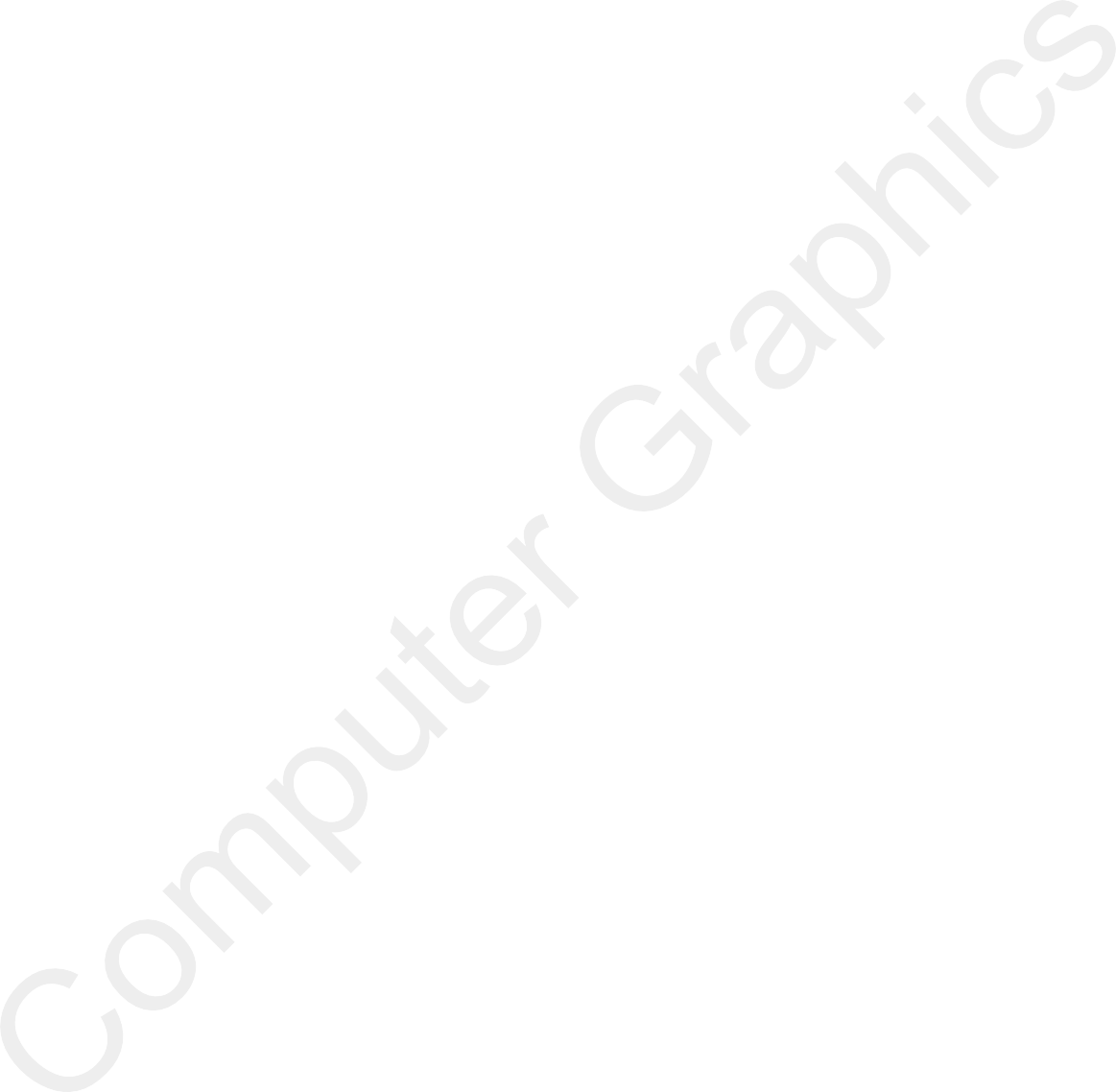
delay(500);

break;

case 2: p3=resetendpt(p1,p2);

p4=resetendpt(p2,p1); drawwindow(); delay(500); drawline(p3,p4); break;

}

delay(5000); closegraph();

}

void drawwindow()

{

line(150,100,450,100); line(450,100,450,350); line(450,350,150,350); line(150,350,150,100);

}

void drawline(PT p1,PT p2)

{

line(p1.x,p1.y,p2.x,p2.y);

}

PT setcode(PT p) //for setting the 4 bit code

{

PT ptemp;

if(p.y<100)

ptemp.code[0]='1'; //Top

else

ptemp.code[0]='0';

if(p.y>350)

ptemp.code[1]='1'; //Bottom

else

ptemp.code[1]='0';

if(p.x>450)

ptemp.code[2]='1'; //Right

else

ptemp.code[2]='0';

if(p.x<150)

ptemp.code[3]='1'; //Left

else

ptemp.code[3]='0';

ptemp.x=p.x; ptemp.y=p.y;

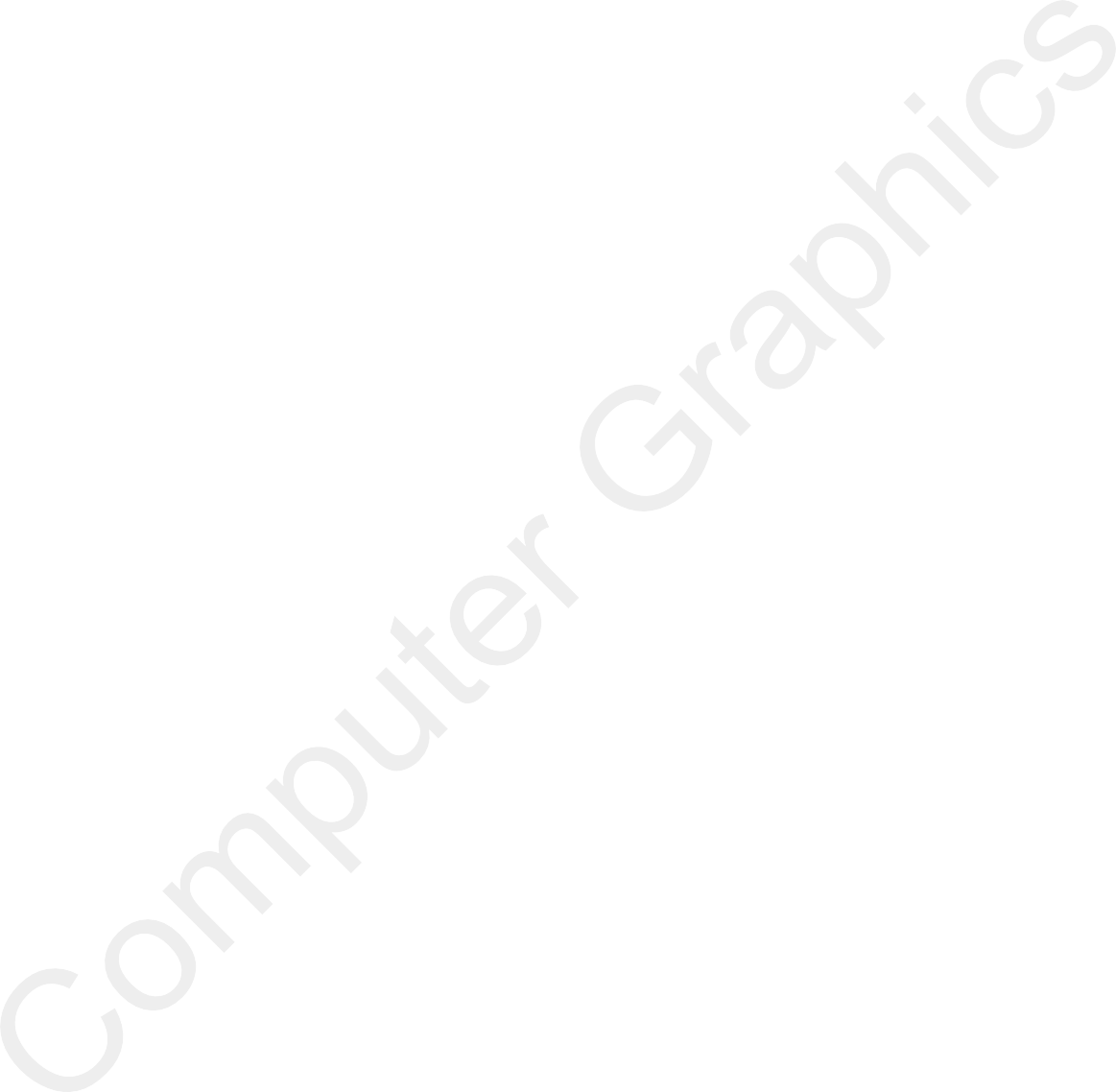
return(ptemp);

}

int visibility(PT p1,PT p2)

{

int i,flag=0;

for(i=0;i<4;i++)

{

if((p1.code[i]!='0') || (p2.code[i]!='0')) flag=1;

}

if(flag==0)

return(0);

for(i=0;i<4;i++)

{

if((p1.code[i]==p2.code[i]) && (p1.code[i]=='1')) flag='0';

}

if(flag==0)

return(1);

return(2);

}

PT resetendpt(PT p1,PT p2)

{

PT temp; int x,y,i; float m,k;

if(p1.code[3]=='1')

x=150;

if(p1.code[2]=='1')

x=450;

if((p1.code[3]=='1') || (p1.code[2]=='1'))

{

m=(float)(p2.y-p1.y)/(p2.x-p1.x);

k=(p1.y+(m\*(x-p1.x))); temp.y=k;

temp.x=x;

for(i=0;i<4;i++)

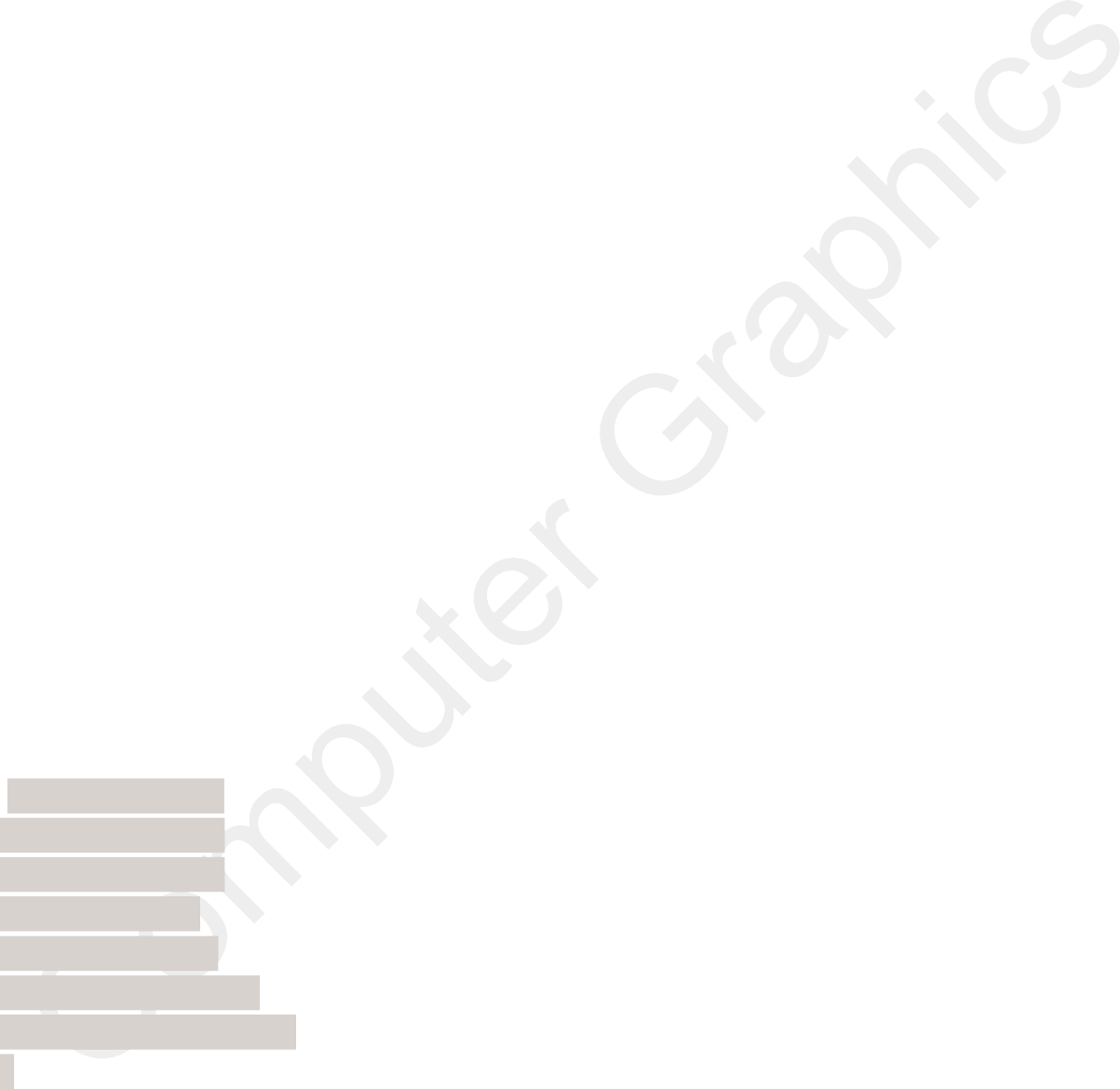
temp.code[i]=p1.code[i];

if(temp.y<=350 && temp.y>=100) return (temp);

}

if(p1.code[0]=='1')

y=100;

if(p1.code[1]=='1')

y=350;

if((p1.code[0]=='1') || (p1.code[1]=='1'))

{

m=(float)(p2.y-p1.y)/(p2.x-p1.x);

k=(float)p1.x+(float)(y-p1.y)/m; temp.x=k;

temp.y=y;

for(i=0;i<4;i++)

temp.code[i]=p1.code[i];

}

else

}

return(temp); return(p1);

# Que(10):Write a Program for Midpoint line clipping Subdivision.

**Ans:-**

#include “stdio.h” #include “conio.h” #include “stdlib.h” #include “dos.h” #include “math.h” #include “graphics.h” typedef struct coordinate

{

int x,y;

char code[4];

}PT;

void drawwindow();

void drawline (PT p1,PT p2); PT setcode(PT p);

int visibility (PT p1,PT p2);

PT resetendpt (PT p1,PT p2); main()

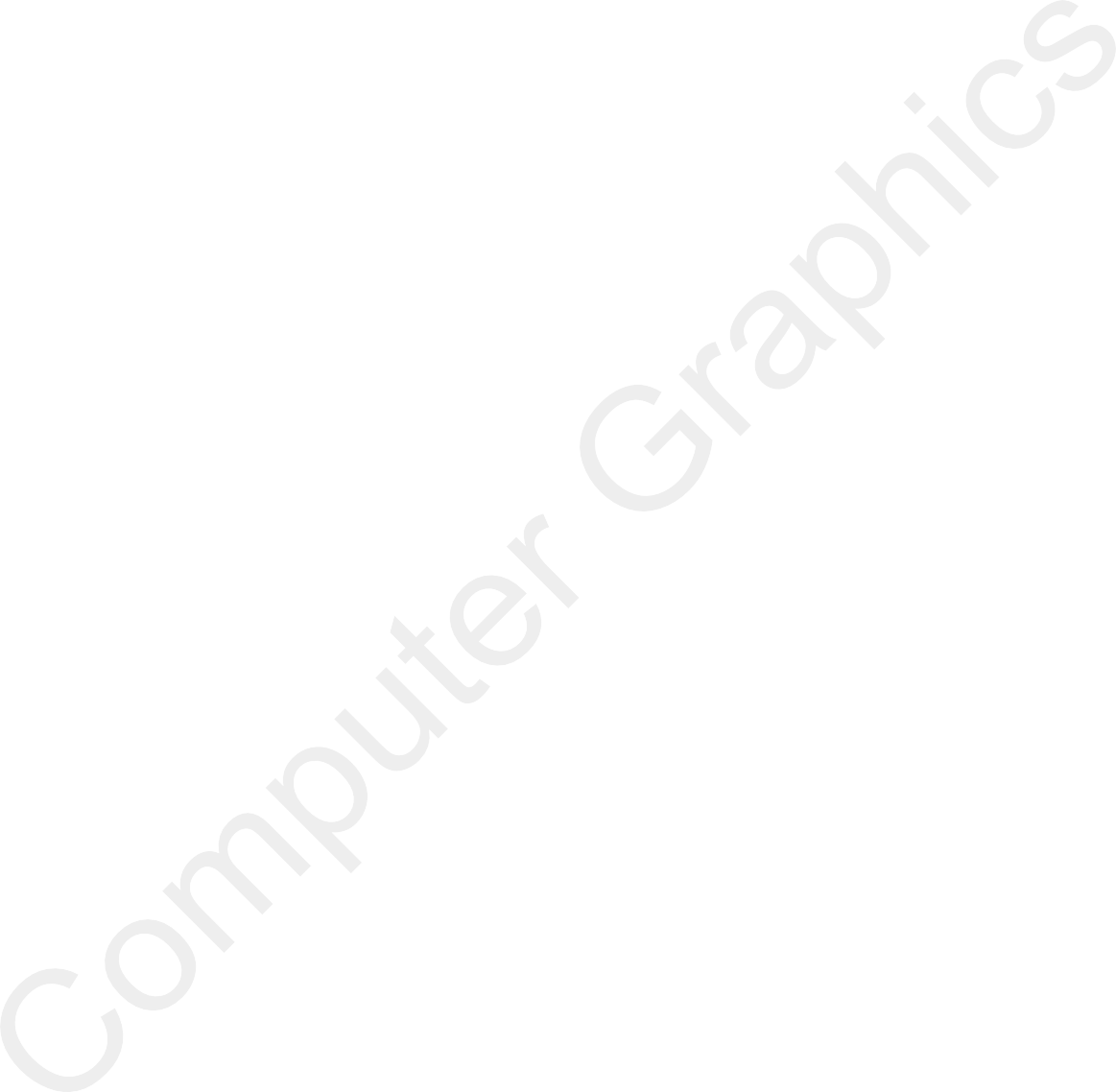
{

int gd=DETECT, gm,v;

PT p1,p2,ptemp;

initgraph(&gd,&gm,”c:\\tc\\bgi”); cleardevice();

printf(“ENTER END-POINT 1 (x,y): “);

scanf(“%d%d”,&p1.x,&p1.y); printf(“\nENTER END-POINT 2 (x,y): “);

scanf(“%d%d”,&p2.x,&p2.y); cleardevice();

drawwindow(); getch(); drawline(p1,p2); getch(); cleardevice(); drawwindow(); midsub(p1,p2); getch(); closegraph(); return(0);

}midsub(PT p1,PT p2)

{

PT mid; int v;

p1=setcode(p1); p2=setcode(p2); v=visibility(p1,p2); switch(v)

{

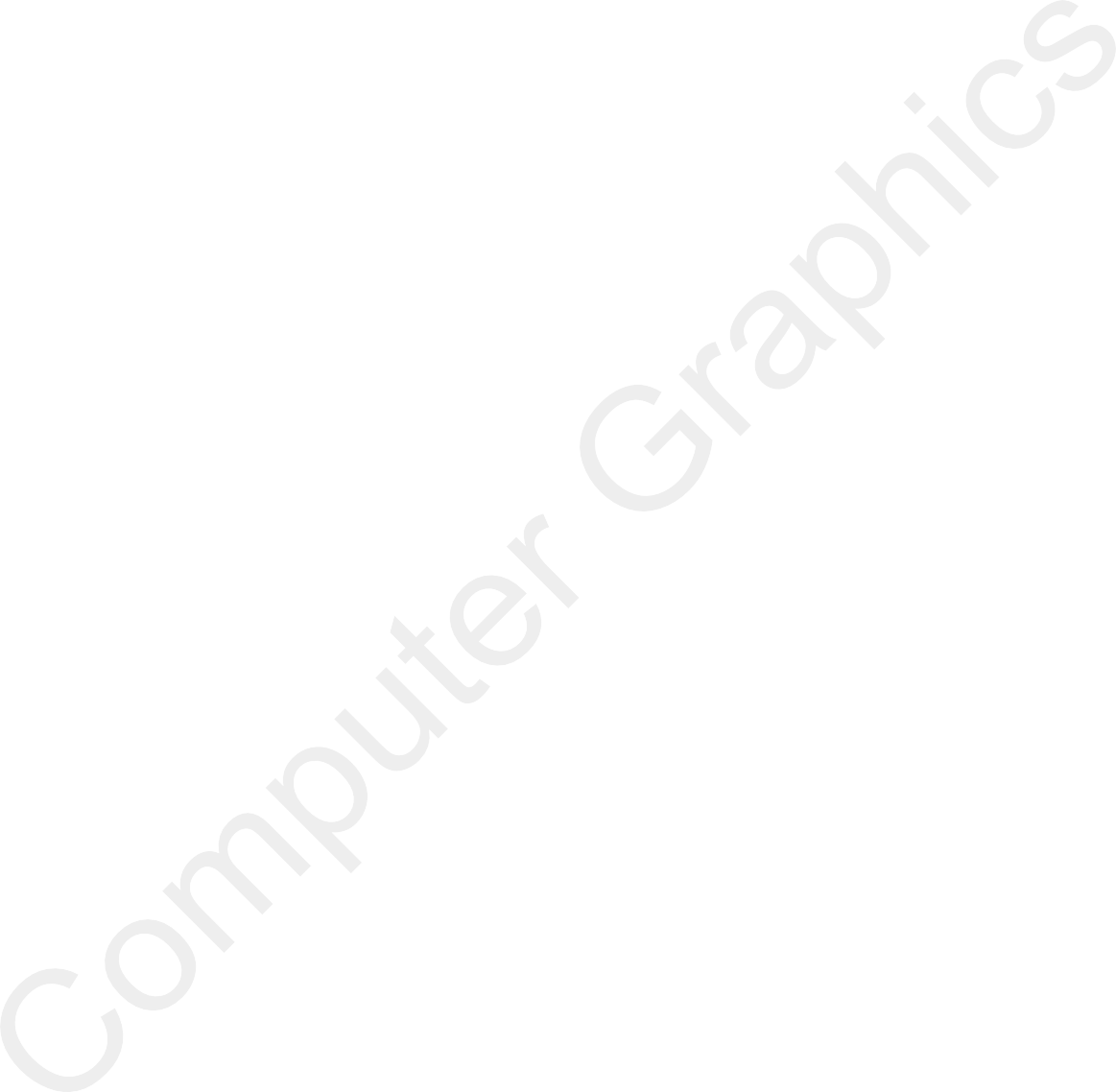
case 0: drawline(p1,p2); break; case 1:break; case 2:

mid.x = p1.x + (p2.x-p1.x)/2;

mid.y = p1.y + (p2.y-p1.y)/2; midsub(p1,mid);

mid.x = mid.x+1; mid.y = mid.y+1; midsub(mid,p2); break;

}



}

void drawwindow()

{

setcolor(RED); line(150,100,450,100); line(450,100,450,400); line(450,400,150,400); line(150,400,150,100);

}

void drawline (PT p1,PT p2)

{

setcolor(15); line(p1.x,p1.y,p2.x,p2.y);

}

PT setcode(PT p)

{

PT ptemp; if(p.y<=100) ptemp.code[0]=’1′; else ptemp.code[0]=’0′; if(p.y>=400) ptemp.code[1]=’1′;

else ptemp.code[1]=’0′; if (p.x>=450) ptemp.code[2]=’1′; else ptemp.code[2]=’0′; if (p.x<=150) ptemp.code[3]=’1′; else ptemp.code[3]=’0′; ptemp.x=p.x; ptemp.y=p.y; return(ptemp);

}

int visibility (PT p1,PT p2)

{

int i,flag=0; for(i=0;i<4;i++)

{

if((p1.code[i]!=’0′)||(p2.code[i]!=’0′)) flag=1;

}

if(flag==0) return(0); for(i=0;i<4;i++)

{

if((p1.code[i]==p2.code[i]) &&(p1.code[i]==’1′)) flag=0;

}

if(flag==0) return(1); return(2);

}

