Exp 1(DDA)

#include<stdio.h>

#include<conio.h>

#include<graphics.h>

void main()

{

int x1,y1,x2,y2,dx,dy,step,i;

float x,y,xinc,yinc;

int gd=DETECT,gm;

//clrscr();

//gd=DETECT;

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

printf("Enter co-ordinate of first point:");

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

printf("Enter co-ordinate of second point:");

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

dx=x2-x1;

dy=y2-y1;

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

step=abs(dx);

else

step=abs(dy);

xinc=dx/(float)step;

yinc=dy/(float)step;

x=x1;

y=y1;

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

{

putpixel(x,y,10);

x=x+xinc;

y=y+yinc;

delay(10);

}

getch();

closegraph();

}

Exp 2 (bresenhams)

#include<stdio.h>

#include<conio.h>

#include<graphics.h>

void main()

{

int x1,y1,x2,y2,dx,dy,step,i;

float x,y,xinc,yinc;

int gd=DETECT,gm;

//clrscr();

//gd=DETECT;

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

printf("Enter co-ordinate of first point:");

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

printf("Enter co-ordinate of second point:");

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

dx=x2-x1;

dy=y2-y1;

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

step=abs(dx);

else

step=abs(dy);

xinc=dx/(float)step;

yinc=dy/(float)step;

x=x1;

y=y1;

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

{

putpixel(x,y,10);

x=x+xinc;

y=y+yinc;

delay(10);

}

getch();

closegraph();

}

Exp 3(midpoint circle)

#include<stdio.h>

#include<graphics.h>

#include<conio.h>

#include<math.h>

void main()

{

int gdriver=DETECT, gmode;

intigraph(&gdriver,&gmode,"C:\\t

c\\BGI");

int xc, yc, x, y;

float p, r;

clrscr();

intigraph(&gdriver,&gmode,"C:\\t

c\\BGI");

printf("Enter the centre

coordinates:");

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

printf("\n Enter the radius:");

scanf("%f",&r);

x=0;

y=r;

p=(5/4)-r;

do{

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

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

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

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

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

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

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

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

if(p<0)

{

p=p+((2\*x)+3);

}

else{

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

y--;

}

x++;

}

while(x<y);

getch();

closegraph();

}

Exp 4(midpoint ellipse)

| #include<stdio.h>  #include<conio.h>  #include<graphics.h>  #include<math.h>  void midptellipse(int rx,int ry,int xc,int yc) {  float dx,dy,d1,d2,x,y;  x=0; y=ry;  d1=(ry\*ry)-(rx\*rx\*ry)+(0.25\*rx\*rx);  dx=2\*ry\*ry\*x;  dy=2\*rx\*rx\*y;  while(dx<dy)  {  putpixel(x+xc,y+yc,WHITE);  putpixel(-x+xc,y+yc,WHITE);  putpixel(x+xc,-y+yc,WHITE);  putpixel(-x+xc,-y+yc,WHITE);  if(d1<0){  x++;  dx=dx+(2\*ry\*ry);  d1=d1+dx+(ry\*ry);  }  else{  x++;  y--;  dx=dx+(2\*ry\*ry);  dy=dy-(2\*rx\*rx);  d1=d1+dx-dy+(ry\*ry);  }  }  d2=((ry\*ry)\*((x+0.5)\*(x+0.5)))+((rx\*rx)\*((y-1)\*(y-1)))-(rx\*rx\*ry\*ry);  while(y>0){  putpixel(x+xc,y+yc,WHITE);  putpixel(-x+xc,y+yc,WHITE);  putpixel(x+xc,-y+yc,WHITE);  putpixel(-x+xc,-y+yc,WHITE);  if(d2>0){  y--;  dy=dy-(2\*rx\*rx);  CSL303- Computer Graphics Lab  Vidyavardhini’s College of Engineering & Technology  Department of Computer Science & Engineering (Data  Science)  d2=d2+(rx\*rx)-dy;  }  else{  y--;  x++;  dx=dx+(2\*ry\*ry);  dy=dy-(2\*rx\*rx);  d2=d2+dx-dy+(rx\*rx);  }  }  }  int main()  {  int gd=DETECT,gm;  initgraph(&gd,&gm,"C:\\tc\\BGI");  midptellipse(10,15,50,50);getch();  closegraph();  return 0;  } |
| --- |

Exp 5(boundary fill ,flood fil)

**BOUNDARY FILL**

#include<stdio.h>

#include<conio.h>

#include<graphics.h>

#include<math.h>

void boundryfill(int x,int y,int fill,int boundry) {

int current;

current=getpixel(x,y);

if((current!=boundry)&&(current!=fill)) {

setcolor(fill);

putpixel(x,y,fill);

delay(3);

boundryfill(x+1,y,fill,boundry);

boundryfill(x-1,y,fill,boundry);

boundryfill(x,y+1,fill,boundry);

boundryfill(x,y-1,fill,boundry);

}

}

void main()

{

int gd=DETECT,gm;

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

setcolor(10);

rectangle(250,200,310,260);

boundryfill(280,250,12,10);

getch();

}

**FLOOD FILL**

#include<stdio.h>

#include<conio.h>

#include<graphics.h>

#include<math.h>

void flood(int x,int y,int fillcolor,int oldcolor)

{

int current;

current=getpixel(x,y);

if(current==oldcolor)

{

delay(3);

putpixel(x,y,fillcolor);

flood(x+1,y,fillcolor,oldcolor);

flood(x-1,y,fillcolor,oldcolor);

flood(x,y+1,fillcolor,oldcolor);

flood(x,y-1,fillcolor,oldcolor);

}

}

void main()

{

int gd=DETECT,gm;

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

setcolor(GREEN);

line(50,50,100,50);

line(50,50,50,75);

line(100,50,100,75);

setcolor(RED);

line(50,75,50,100);

line(50,100,100,100);

line(100,75,100,100);

flood(55,55,YELLOWCS,0);

getch();

closegraph();

}

Exp 6(translation)

#include<stdio.h>

#include<conio.h>

#include<graphics.h>

#include<math.h>

void midptellipse(int rx,int ry,int xc,int yc) {

float dx,dy,d1,d2,x,y;

x=0; y=ry;

d1=(ry\*ry)-(rx\*rx\*ry)+(0.25\*rx\*rx);

dx=2\*ry\*ry\*x;

dy=2\*rx\*rx\*y;

while(dx<dy)

{

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

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

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

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

if(d1<0){

x++;

dx=dx+(2\*ry\*ry);

d1=d1+dx+(ry\*ry);

}

else{

x++;

y--;

dx=dx+(2\*ry\*ry);

dy=dy-(2\*rx\*rx);

d1=d1+dx-dy+(ry\*ry);

}

}

d2=((ry\*ry)\*((x+0.5)\*(x+0.5)))+((rx\*rx)\*((y-1)\*(y-1)))-(rx\*rx\*ry\*ry);

while(y>0){

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

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

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

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

if(d2>0){

y--;

dy=dy-(2\*rx\*rx);

d2=d2+(rx\*rx)-dy;

}

else{

y--;

x++;

dx=dx+(2\*ry\*ry);

dy=dy-(2\*rx\*rx);

d2=d2+dx-dy+(rx\*rx);

}

}

}

int main()

{

int gd=DETECT,gm;

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

midptellipse(10,15,50,50);getch();

closegraph();

return 0;

}

Exp 7(cohensutherland)

#include<conio.h>

#include<stdio.h>

#include<graphics.h>

#include<math.h>

void main()

{

int a[4],b[4];

float m,xnew,ynew;

float x1=100,y1=100,xh=300,yh=300,xa=10,ya=200,xb=250,yb=150;

int gd=DETECT,gm;

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

setcolor(5);

line(xa,ya,xb,yb);

setcolor(12);

rectangle(x1,y1,xh,yh);

m = (yb-ya)/(xb-xa);

if(xa<x1)

a[3]=1;

else a[3]=0;

if(xa>xh)

a[2]=1;

else a[2]=0;

if(ya<y1)

a[1]=1;

else a[1]=0;

if(ya>yh)

a[0]=1;

else a[0]=0;

if(xb<x1)

b[3]=1;

else b[3]=0;

if(xb>xh)

b[2]=1;

else b[2]=0;

if(yb<y1)

b[1]=1;

else b[1]=0;

if(yb>yh)

b[0]=1;

else b[0]=0;

printf("press a key to continue");

getch();

if(a[0] == 0 && a[1] == 0 && a[2] == 0 && a[3] == 0 && b[0] == 0 && b[1] == 0 && b[2] == 0 && b[3] == 0)

{

printf("no clipping");

line(xa,xb,ya,yb);

}

else if(a[0]&&b[0] || a[1]&&b[1] || a[2]&&b[2] || a[3]&&b[3])

{

clrscr();

printf("line discarded");

rectangle(x1,y1,xh,yh);

}

else

{

if(a[3] == 1 && b[3] == 0)

{

ynew = (m\*(x1-xa)) + ya;

setcolor(12);

rectangle(x1,y1,xh,yh);

setcolor(0);

line(xa,ya,xb,yb);

setcolor(15);

line(x1,ynew,xb,yb);

}

else if(a[2] == 1 && b[2] == 0)

{

xnew = xa + (y1-ya)/m;

setcolor(0);

line(xa,ya,xb,yb);

setcolor(15);

line(xnew,yh,xb,yb);

}

else if(a[0] == 1 && b[0] == 0)

{

xnew = xa + (yh-ya)/m;

setcolor(0);

line(xa,ya,xb,yb);

setcolor(15);

line(xnew,yh,xb,yb);

}

}

getch();

closegraph();

}

Exp 8(Sutherland Hodgeman )

#include <stdio.h>

#include <graphics.h>

#include <conio.h>

#include <stdlib.h>

int main()

{

int gd, gm, n, \*x, i, k = 0;

// window coordinates int

wx1 = 220, wy1 = 140, wx2 = 420, wy2 = 140, wx3 = 420, wy3 = 340, wx4 = 220, wy4 = 40;

int w[] = {220, 140, 420, 140, 420, 340, 220, 340, 220, 140}; // array for drawing window

detectgraph(&gd, &gm);

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

printf("Window:-");

setcolor(RED); // red colored window

drawpoly(5, w); // window drawn

printf("Enter the no. of vertices of polygon: ");

scanf("%d", &n);

x = malloc(n \* 2 + 1);

printf("Enter the coordinates of points:\n");

k = 0;

for (i = 0; i < n \* 2; i += 2) // reading vertices of polygon

{

printf("(x%d,y%d): ", k, k);

scanf("%d,%d", &x[i], &x[i + 1]);

k++;

}

x[n \* 2] = x[0]; // assigning the coordinates of first vertex to last additional vertex for drawpoly method.

x[n \* 2 + 1] = x[1];

setcolor(WHITE);

drawpoly(n + 1, x);

printf("\nPress a button to clip a polygpolygon..");

getch();

setcolor(RED);

drawpoly(5, w);

setfillstyle(SOLID\_FILL, BLACK);

floodfill(2, 2, RED);

gotoxy(1, 1); // bringing cursor at starting position

printf("\nThis is the clipped polygon..");

getch();

cleardevice();

closegraph();

return 0;

}

Exp 9(beizier curve)

#include <graphics.h>

#include <math.h>

#include <conio.h>

#include <stdio.h>

void main() {

int x[4], y[4], i;

double put\_x, put\_y, t;

int gr = DETECT, gm;

initgraph(&gr, &gm, "C:\\tc\\BGI");

printf("\n\*\*\* Bezier Curve \*\*\*\*");

printf("\n Please enter x and y coordinates (4 control points):\n");

// Get control points from user

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

printf("Control Point %d: ", i + 1);

scanf("%d%d", &x[i], &y[i]);

putpixel(x[i], y[i], 3); // Draw control points

}

// Draw Bézier curve

for (t = 0.0; t <= 1.0; t += 0.001) { // Increment t in small steps

put\_x = pow(1 - t, 3) \* x[0] +

3 \* t \* pow(1 - t, 2) \* x[1] +

3 \* t \* t \* (1 - t) \* x[2] +

pow(t, 3) \* x[3]; // X coordinate of the Bézier curve

put\_y = pow(1 - t, 3) \* y[0] +

3 \* t \* pow(1 - t, 2) \* y[1] +

3 \* t \* t \* (1 - t) \* y[2] +

pow(t, 3) \* y[3]; // Y coordinate of the Bézier curve

putpixel((int)put\_x, (int)put\_y, WHITE); // Draw the curve

}

getch(); // Wait for a key press

closegraph(); // Close the graphics window

}

Exp 10(animation)

#include <stdio.h>

#include <stdlib.h>

#include <graphics.h>

#include <conio.h>

void main(){

int

gdriver=DETECT,gmode,col=480,row=640,font=4,direction=2,

size=8,color=15;

initgraph(&gdriver,&gmode,"C:\\tc\\BGI");

cleardevice();

while(!kbhit()){

settextstyle(random(font),random(direction

),random(size)); setcolor(random(color));

outtextxy(random(col),random(row),"Manaswi");

delay(250);

}

closegraph();

}