Practical 1b

Draw co-orinate axis at the center

#include<stdio.h>

#include<conio.h>

#include<graphics.h>

void main()

{

int gd=DETECT,gm;

int midx,midy;

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

cleardevice();

midx=getmaxx()/2;

midy=getmaxy()/2;

line(1,midy,640,midy);

line(midx,1,midx,480);

outtextxy(50,300,"Name");

getch();

}

Practical 2a

Draw circle,rectangle,elipse,half-elipse

#include<stdio.h>

#include<conio.h>

#include<graphics.h>

void main()

{

int gd=DETECT,gm;

int midx,midy;

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

cleardevice();

midx=getmaxx()/2;

midy=getmaxy()/2;

line(1,midy,640,midy);

line(midx,1,midx,480);

setcolor(WHITE);

circle(midx+(-150),midy-(120),40);

printf("\t\tIt's a circle");

setcolor(GREEN);

rectangle(midx+(100),midy-(100),midx+(200),midy-(150));

printf("\t\t\t\tThis is rectangle\n\n\n\n\n");

setcolor(BLUE);

ellipse(midx+(-150),midy-(-100),0,360,midx+(-250),midy-(200));

printf("\n\n\n\n\n\n\n\n\n\n\n\t wow!! It's an ellipse");

setcolor(YELLOW);

ellipse(midx+(180),midy-(-100),180,0,midx+(-200),midy-(150));

printf("\t\t\t ohhh!!half ellipse! Looks like:");

outtextxy(100,50,"Name");

getch();

}

Practical 2b1

Simple hut

#include<stdio.h>

#include<conio.h>

#include<graphics.h>

void main()

{

int gd=DETECT,gm;

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

setcolor(WHITE);

rectangle(150,180,250,300);

rectangle(250,180,420,300);

rectangle(180,250,220,300);

outtextxy(350,200,"Name");

line(200,100,150,180);

line(200,100,250,180);

line(200,100,370,100);

line(370,100,420,180);

getch();

closegraph();

}

Practical 2b2

Colourfull hut

#include<graphics.h>

#include<stdio.h>

#include<conio.h>

void main()

{

int gd=DETECT,gm;

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

setcolor(WHITE);

rectangle(150,180,250,300);

rectangle(250,180,420,300);

rectangle(180,250,220,300);

outtextxy(350,200,"Name");

line(200,100,150,180);

line(200,100,250,180);

line(200,100,370,100);

line(370,100,420,180);

setfillstyle(SOLID\_FILL,BROWN);

floodfill(152,182,WHITE);

floodfill(252,182,WHITE);

setfillstyle(SLASH\_FILL,BLUE);

floodfill(182,252,WHITE);

setfillstyle(HATCH\_FILL,GREEN);

floodfill(200,105,WHITE);

floodfill(210,105,WHITE);

getch();

closegraph();

}

Practical 3

Basic shapes(circle,rectangle,square,concentric circle,ellipse line)

#include<stdio.h>

#include<conio.h>

#include<graphics.h>

void main()

{

int gd=DETECT,gm,left=100,top=100,right=200,bottom=200,x=300,y=150,radius=50;

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

rectangle(120,150,230,200);

circle(x,y,radius);

bar(left+300,top,right+300,bottom);

line(left-10,top+150,left+410,top+150);

ellipse(x,y+200,0,360,80,50);

for(radius=25;radius<=100;radius=radius+20)

circle(500,350,radius);

outtextxy(275,150,"Name");

getch();

closegraph();

}

Practical 4a

DDA line algorithm

#include<stdio.h>

#include<conio.h>

#include<math.h>

#include<dos.h>

#include<graphics.h>

void main()

{

float x,y,x1,y1,x2,y2,dx,dy,pixel;

int i,gd,gm;

printf("Enter the value of x1:");

scanf("%f",&x1);

printf("Enter the value of y1:");

scanf("%f",&y1);

printf("Enter the value of x2:");

scanf("%f",&x2);

printf("Enter the value of y2:");

scanf("%f",&y2);

detectgraph(&gd,&gm);

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

dx=abs(x2-x1);

dy=abs(y2-y1);

if(dx>=dy)

pixel=dx;

else

pixel=dy;

dx=dx/pixel;

dy=dy/pixel;

x=x1;

y=y1;

i=1;

while(i<=pixel)

{

putpixel(x,y,1);

x=x+dx;

y=y+dy;

i=i+1;

delay(100);

}

getch();

closegraph();

}

o/p:x1=100,x2=200,x3=300,x4=400

Practical 4b

Bresenhams line drawing algorithm

#include<stdio.h>

#include<conio.h>

#include<graphics.h>

void main()

{

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

int gd,gm;

clrscr();

printf("\n\n\tEnter the co-ordinates of first point:");

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

printf("\n\n\tEnter the co-ordinates of second point:");

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

dx=(x2-x1);

dy=(y2-y1);

p=2\*(dy)-(dx);

x=x1;

y=y1;

detectgraph(&gd,&gm);

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

putpixel(x,y,WHITE);

while(x<=x2)

{

if(p<0)

{

x=x+1;

y=y;

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

}

else

{

x=x+1;

y=y+1;

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

}

putpixel(x,y,WHITE);

}

getch();

closegraph();

}

o/p:First point:100 200 , 300 400

Practical 5a

Midpoint circle drawing algo

#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("\*\*\*Bresenham's Midpoint 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)

{

x=x+1;

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

}

else

{

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);

}

o/p:Xc=100,Yc=200,radius of circle=50

Practical 5b

Mid point ellipse drawing algo

#include<conio.h>

#include<stdio.h>

#include<graphics.h>

#include<math.h>

void disp();

float x,y;

int xc,yc;

void main()

{

int gd=DETECT,gm;

int a,b;

float p1,p2;

clrscr();

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

printf("Enter xc:\t");

scanf("%d",&xc);

printf("Enter yc:\t");

scanf("%d",&yc);

printf("Enter a:\t");

scanf("%d",&a);

printf("Enter b:\t");

scanf("%d",&b);

x=0;y=b;

disp();

p1=(b\*b)-(a\*a\*b)+(a\*a)/4;

while((2.0\*b\*b\*x)<=(2.0\*a\*a\*y))

{

x++;

if(p1<=0)

p1=p1+(2.0\*b\*b\*x)+(b\*b);

else

{

y--;

p1=p1+(2.0\*b\*b\*x)+(b\*b)-(2.0\*a\*a\*y);

}

disp();

x=-x;

disp();

x=-x;

}

x=a;

y=0;

disp();

p2=(a\*a)+2.0\*(b\*b\*a)+(b\*b)/4 ;

while((2.0\*b\*b\*x)>(2.0\*a\*a\*y))

{

y++;

if(p2>0)

p2=p2+(a\*a)-(2.0\*a\*a\*y);

else

{

x--;

p2=p2+(2.0\*b\*b\*x)-(2.0\*a\*a\*y)+(a\*a);

}

disp();

y=-y;

disp();

y=-y;

}

getch();

closegraph();

}

void disp()

{

putpixel(xc+x,yc+y,10);

putpixel(xc-x,yc+y,10);

putpixel(xc+x,yc-y,10);

putpixel(xc+x,yc-y,10);

}

o/p:Xc=100,Yc=200,a=30,b=60

Practical 6a

Implementing 2D scaling

#include<stdio.h>

#include<conio.h>

#include<graphics.h>

void main()

{

int i;

int gd=DETECT,gm;

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

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

printf("Enter the 2 line end points:x1,y1,x2,y2:\n");

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

line(x1,y1,x2,y2);

printf("\nEnter scaling co-ordinates;x\t y\t \n");

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

x1=(x1\*x);

y1=(y1\*y);

x2=(x2\*x);

y2=(y2\*y);

printf("Line after scaling");

line(x1,y1,x2,y2);

getch();

closegraph();

}

0/p:x1,y1,x2,y2=30,40,50,60 co-ordinates=5 6

Practical 6b

2D translation

#include<stdio.h>

#include<conio.h>

#include<graphics.h>

void main()

{

int i;

int gd=DETECT,gm;

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

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

printf("Enter the 2 line end points:x1,y1,x2,y2:\n");

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

line(x1,y1,x2,y2);

printf("\nEnter scaling co-ordinates;x\t y\t \n");

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

x1=(x1+x);

y1=(y1+y);

x2=(x2+x);

y2=(y2+y);

printf("Line after scaling");

line(x1,y1,x2,y2);

getch();

closegraph();

}

o/p:x1,y1,x2,y2=30,40,50,60 co-ordinates=100 200

Practical 7

Perform 2D rotation on given object

#include<graphics.h>

#include<stdio.h>

void main()

{

int i;

int gd=DETECT,gm;

int x2,y2,x1,y1,x,y,xn,yn;

double r11,r12,r21,r22,th;

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

printf("Enter the 2 line end points:x1,y1,x2,y2:\n");

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

line(x1,y1,x2,y2);

printf("\n\n\nEnter the angle:\t");

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\*r21));

yn=((x2\*r12)-(y2\*r22));

line(x1,y1,xn,yn);

getch();

closegraph();

}

o/p: x1,y1,x2,y2=100 150 40 30 , Angle=45

Practical 9a

To fill circle using floodfill algo

#include<graphics.h>

#include<stdio.h>

void floodFill(int x,int y,int oldcolor,int newcolor)

{

if(getpixel(x,y)==oldcolor)

{

putpixel(x,y,newcolor);

floodFill(x+1,y,oldcolor,newcolor);

floodFill(x,y+1,oldcolor,newcolor);

floodFill(x-1,y,oldcolor,newcolor);

floodFill(x,y-1,oldcolor,newcolor);

}

}

void main()

{

int gm,gd=DETECT,radius;

int x,y;

printf("Enter x and y position for circle\n");

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

printf("Enter radius of circle\n");

scanf("%d",&radius);

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

circle(x,y,radius);

floodFill(x,y,0,15);

delay(5000);

closegraph();

}

o/p: x&y=100 100 , radius=25

Practical 9b

To fill circle using boundaryfill algo

#include<graphics.h>

#include<stdio.h>

void boundaryFill(int x,int y,int f\_color,int b\_color)

{

if(getpixel(x,y)!=b\_color && getpixel(x,y)!=f\_color)

{

putpixel(x,y,f\_color);

boundaryFill(x+1,y,f\_color,b\_color);

boundaryFill(x,y+1,f\_color,b\_color);

boundaryFill(x-1,y,f\_color,b\_color);

boundaryFill(x,y-1,f\_color,b\_color);

}

}

void main()

{

int gm,gd=DETECT,radius;

int x,y;

printf("Enter x and y position for circle\n");

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

printf("Enter radius of circle\n");

scanf("%d",&radius);

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

circle(x,y,radius);

boundaryFill(x,y,4,15);

delay(5000);

closegraph();

}

o/p: x&y=100 100 , radius=25

Practical 10a

Simple text screen saver using graphic function

#include<graphics.h>

#include<stdio.h>

#include<stdlib.h>

#include<conio.h>

void main()

{

int gm,gd=DETECT,x=600,i;

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

for(x=0;x<250;x++)

{

x%=250;

setcolor(random(16));

circle(random(635),random(70),50);

circle(random(635),random(70),50);

circle(random(635),random(70),50);

circle(random(635),random(70),50);

circle(random(635),random(70),50);

clearviewport();

settextstyle(1,0,5);

setcolor(RED);

outtextxy(50,415-2\*x,"World");

setcolor(GREEN);

outtextxy(200,415-2\*x,"of");

setcolor(YELLOW);

settextstyle(3,0,5);

outtextxy(350,415-2\*x,"graphic");

}

getch();

}

Practical 10b

Smilling face animation

#include<graphics.h>

#include<stdio.h>

#include<conio.h>

void main()

{

int gm,gd=DETECT,x=600,i;

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

}

Practical 10c

Moving car on screen

#include<conio.h>

#include<stdio.h>

#include<graphics.h>

#include<dos.h>

void main()

{

int gd=DETECT,gm;

int I,maxx,midy;

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

maxx=getmaxx();

midy=getmaxy()/2;

for(I=0;I<maxx-150;I=I+5)

{

cleardevice();

setcolor(WHITE);

line(0,midy+37,maxx,midy+37);

setcolor(YELLOW);

setfillstyle(SOLID\_FILL,RED);

line(I,midy+23,I,midy);

line(I,midy,40+I,midy-20);

line(40+I,midy-20,80+I,midy-20);

line(80+I,midy-20,100+I,midy);

line(100+I,midy,120+I,midy);

line(120+I,midy,120+I,midy+23);

line(0+I,midy+23,18+I,midy+23);

arc(30+I,midy+23,0,180,12);

line(42+I,midy+23,78+I,midy+23);

arc(90+I,midy+23,0,180,12);

line(102+I,midy+23,120+I,midy+23);

line(28+I,midy,43+I,midy-15);

line(43+I,midy-15,57+I,midy-15);

line(57+I,midy-15,57+I,midy);

line(57+I,midy,28+I,midy);

line(62+I,midy-15,77+I,midy-15);

line(77+I,midy-15,92+I,midy);

line(92+I,midy,62+I,midy);

line(62+I,midy,62+I,midy-15);

floodfill(5+I,midy+22,YELLOW);

setcolor(BLUE);

setfillstyle(SOLID\_FILL,DARKGRAY);

circle(30+I,midy+25,9);

circle(90+I,midy+25,9);

floodfill(30+I,midy+25,BLUE);

floodfill(90+I,midy+25,BLUE);

delay(100);

}

getch();

closegraph();

}