C Program for Bresenham 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,"e:\\tc\\bgi");

putpixel(x,y,WHITE);

while(x <= x2)

{

if(p < 0)

{

x=x+1;

y=y;

p = p + 2 \* (dy);

}

else

{

x=x+1;

y=y+1;

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

}

putpixel(x,y,WHITE);

}

getch();

closegraph();

}

DDA Line Drawing Algorithm

#include<stdio.h>

#include<conio.h>

#include<graphics.h>

#include<ctype.h>

#include<math.h>

#include<stdlib.h>

void draw(int x1,int y1,int x2,int y2);

void main()

{

int x1,y1,x2,y2;

int gdriver=DETECT,gmode,gerror;

initgraph(&gdriver,&gmode,”c:\\tc\\bgi:”);

printf(“\n Enter the x and y value for starting point:\n”);

scanf(“%d%d”,&x1,&y1);

printf(“\n Enter the x and y value for ending point:\n”);

scanf(“%d%d”,&x2,&y2);

printf(“\n The Line is shown below: \n”);

draw(x1,y1,x2,y2);

getch();

}

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

{

float x,y,xinc,yinc,dx,dy;

int k;

int step;

dx=x2-x1;

dy=y2-y1;

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

step=abs(dx);

else

step=abs(dy);

xinc=dx/step;

yinc=dy/step;

x=x1;

y=y1;

putpixel(x,y,1);

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

{

x=x+xinc;

y=y+yinc;

putpixel(x,y,2);

}

}

Program to draw a Circle using Direct Algorithm

**#include<graphics.h>   
#include<iostream.h>   
#include<stdio.h>   
#include<conio.h>   
  
void main()  
{   
    int gd=DETECT, gm;  
    float x1,x2,y1,y2,x,y,m,c,dx,dy,clr;  
    initgraph(&gd,&gm," ");  
    cout<<"enter the value of x1,y1,x2,y2 and color";  
    cin>>x1>>y1>>x2>>y2>>clr;  
    x=x1; y=y1;  
    dx=x2-x1;  
    dy=y2-y1;  
    m=dy/dx;  
    c=y-m\*c;  
      
    if(dx==0)  
    {  
        for(;y<=y2;y++)  
        {  
            x=x1;  
            putpixel(x,y,clr);  
        }  
    }  
    else if(dy==0)  
    {  
        for(;x<=x2;x++)  
        {  
            y=y1;  
            putpixel(x,y,clr);  
        }  
    }  
    else  
    {  
        m=dy/dx;  
        if(m<1)  
        {  
            for(;x<=x2;x++)  
            {  
                y=m\*x+ c;  
                putpixel(x,y,clr);  
            }  
        }  
        else if(m==1)  
        {  
            for(;x<=x2;x++)  
            {  
                y=y+1;  
                putpixel(x,y,clr);  
            }  
        }  
        else  
        {  
            for(;y<=y2;y++)  
            {  
                x=(y-c)/m;  
                putpixel(x,y,clr);  
            }  
        }  
    }  
    getch();  
    closegraph();  
}**

# To draw a circle using trignometric method

#include<iostream.h>  
#include<graphics.h>  
#include<conio.h>  
#include<math.h>  
void put8pixel(int,int,int,int);  
void main()  
{  
int x,y,x1,y1,r,h,k,theta;  
float n=3.14159/180;  
cout<<”\n\t\t\t\\*Trignometric Method to draw a circle\*\n”;  
cout<<”\nenter the x and y coordinates:-\n”;  
cin>>h>>k;  
cout<<”\nenter the radius:-\n”;  
cin>>r;  
int gd=DETECT,gm;  
initgraph(&gd,&gm,”");  
for(theta=0;theta<=45;theta++)  
{  
x1=r\*cos(theta\*n);  
y1=r\*sin(theta\*n);  
x=int(x1+0.5);  
y=int(y1+0.5);  
put8pixel(x,y,h,k);  
}  
outtextxy(115,70,”circle using Trigonometric method”);  
getch();  
closegraph();  
}  
void put8pixel(int x,int y,int h,int k)  
{  
putpixel(x+h,y+k,6);  
putpixel(x+h,-y+k,6);  
putpixel(-x+h,y+k,6);  
putpixel(-x+h,-y+k,6);  
putpixel(y+h,x+k,6);  
putpixel(y+h,-x+k,6);  
putpixel(-y+h,x+k,6);  
putpixel(-y+h,-x+k,6);  
}

**Boundary FILL:**

#include<stdio.h>  
#include<conio.h>  
#include<graphics.h>  
#include<dos.h>  
  
  
  
void fill\_right(x,y)  
int x , y ;  
{  
if((getpixel(x,y) != WHITE)&&(getpixel(x,y) != RED))  
{  
putpixel(x,y,RED);  
fill\_right(++x,y);  
x = x - 1 ;  
fill\_right(x,y-1);  
fill\_right(x,y+1);  
}  
delay(1);  
}  
  
  
void fill\_left(x,y)  
int x , y ;  
{  
if((getpixel(x,y) != WHITE)&&(getpixel(x,y) != RED))  
{  
putpixel(x,y,RED);  
  
fill\_left(--x,y);  
x = x + 1 ;  
fill\_left(x,y-1);  
fill\_left(x,y+1);  
}  
delay(1);  
}  
  
  
void main()  
{  
int x,y,n,i;  
int gd=DETECT,gm;  
clrscr();  
  
initgraph(&gd,&gm,"c:\\tc\\bgi");  
  
  
  
/\*- draw object -\*/  
  
line (50,50,200,50);  
line (200,50,200,300);  
line (200,300,50,300);  
line (50,300,50,50);  
  
/\*- set seed point -\*/  
x = 100; y = 100;  
  
fill\_right(x,y);  
fill\_left(x-1,y);  
  
getch();  
}