Name:Shivam

Write C++/Java program to draw line using DDA and Bresenham‘s algorithm. Inherit pixel class and Use function overloading.

#include<iostream>

#include<graphics.h>

using namespace std;

class pixel

{

public:

float x,y,length,dx,dy;

int p;

};

class pixel1:public pixel

{

public:

void DDA(float,float,float,float);

void bresen(float,float,float,float);

int sign(float);

};

int pixel1::sign(float x)

{

if(x<0)

return -1;

else if(x=0)

return 0;

else

return 1;

}

void pixel1::DDA(float x1,float y1,float x2,float y2)

{

dx=abs(x2-x1);

dy=abs(y2-y1);

if(dx>dy)

length=dx;

else

length=dy;

dx=(x2-x1)/length;

dy=(y2-y1)/length;

x=x1+0.5\*sign(dx);

y=y1+0.5\*sign(dy);

for(int i=0;i<length;i++)

{

x=x+dx;

y=y+dy;

putpixel(x,y,WHITE);

}

}

void pixel1::bresen(float x1,float y1,float x2,float y2)

{

dx=x2-x1;

dy=y2-y1;

x=x1;

y=y1;

p=2\*dy-dx;

while(x<x2)

{

if(p>=0)

{

putpixel(x,y,RED);

y=y+1;

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

}

else

{

putpixel(x,y,RED);

p=p+2\*dy;

}

x=x+1;

}

}

int main()

{

int gd=DETECT,gm;

initgraph(&gd,&gm,NULL);

pixel1 s;

float x1,y1,x2,y2;

char ans;

int ch;

do

{

cout<<"\n\*\*\*\*MENU\*\*\*\*";

cout<<"\n1.DDA ";

cout<<"\n2.Bresenham ";

cout<<"\n3.Exit ";

cin>>ch;

switch(ch)

{

case 1: cleardevice();

cout<<"\nEnter co-ordinates of line(x1,y1,x2,y2)";

cin>>x1>>y1>>x2>>y2;

outtextxy(200,100,"DDA line algorithm : ");

s.DDA(x1,y1,x2,y2);

break;

case 2: cleardevice();

cout<<"\nEnter co-ordinates of line(x1,y1,x2,y2)";

cin>>x1>>y1>>x2>>y2;

outtextxy(200,100,"Bresenham's line algorithm : ");

s.bresen(x1,y1,x2,y2);

break;

case 3:

break;

}

cout<<"\nDo u want to continue...(y/n)";

cin>>ans;

}while(ans=='y');

delay(10);

closegraph();

}

Output: -







