**Lab 1**

WAP in C to implement DDA algorithm for:

1. |m|>1 positive slope
2. |m|>1 negative slope

Source Code:

#include<stdio.h>

#include<conio.h>

#include<graphics.h>

#include<math.h>

int i,dinc;

int main()

{

printf("\*\*\tCompiled By Sanjog Gautam\t\*\*\n");

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

printf("Enter the starting points(x1,y1): ");

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

printf("Enter the starting points(x2,y2): ");

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

dx=x2-x1;

dy=y2-y1;

float m=float(dy)/dx;

dx=fabs(dx);

dy=fabs(dy);

float x=x1; float y=y1;

int gm,gd=DETECT;

initgraph(&gd,&gm,NULL);

if(fabs(m)<1&&m>=0) {//for positive slope |m|<1

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

{

int x\_inc=1;

putpixel(round(x),round(y),WHITE);

delay(10);

x=x+x\_inc;

y=y+m\*x\_inc;

}

}

else if(fabs(m)>1&&m>=0) //for positive slope |m|>1

{

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

{

int y\_inc=1;

putpixel(round(x),round(y),WHITE);

delay(10);

x=x+(1/m);

y=y+y\_inc;

}

}

else if(fabs(m)>1&&m<0) //for negative slope |m|>1

{

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

{

int y\_inc=-1;

putpixel(round(x),round(y),WHITE);

delay(10);

x=x+(1/m);

y=y+y\_inc;

}

}

else

{

for(i=1;i<=dy;i++) //for negative slope |m|<1

{

int x\_inc=-1;

putpixel(round(x),round(y),WHITE);

delay(10);

x=x+x\_inc;

y=y+m\*x\_inc;

}

}

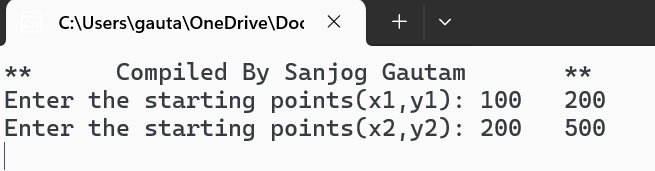
getch();

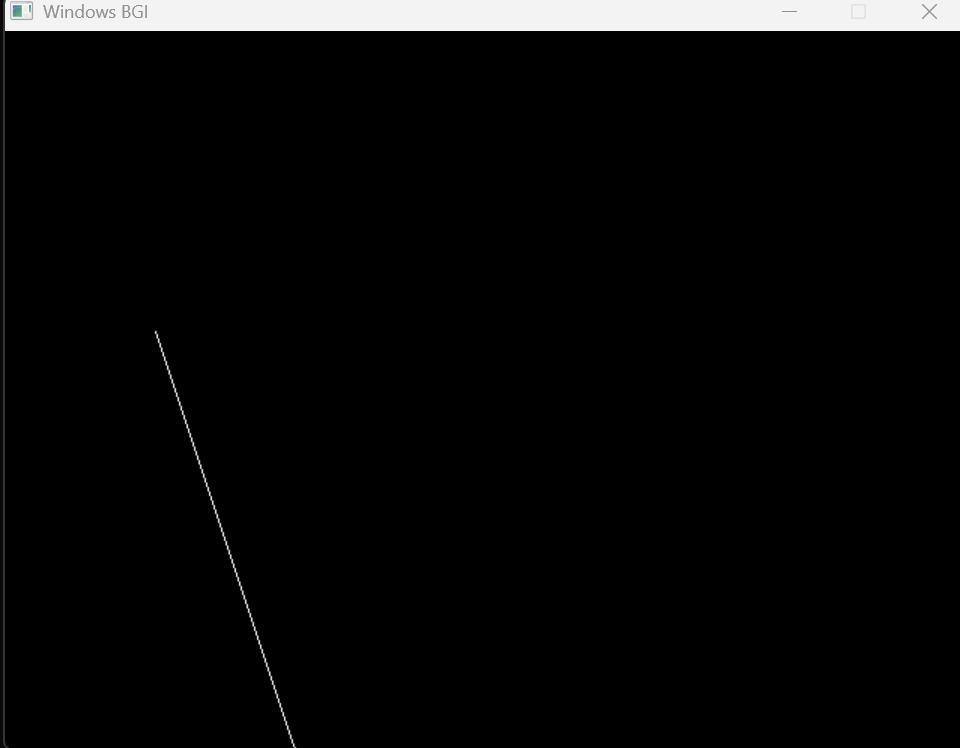
closegraph();

}

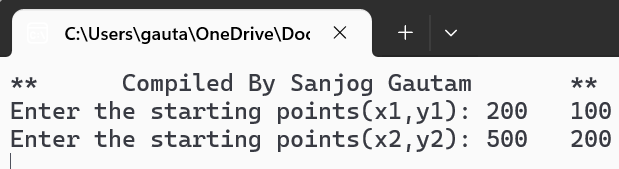
Output:

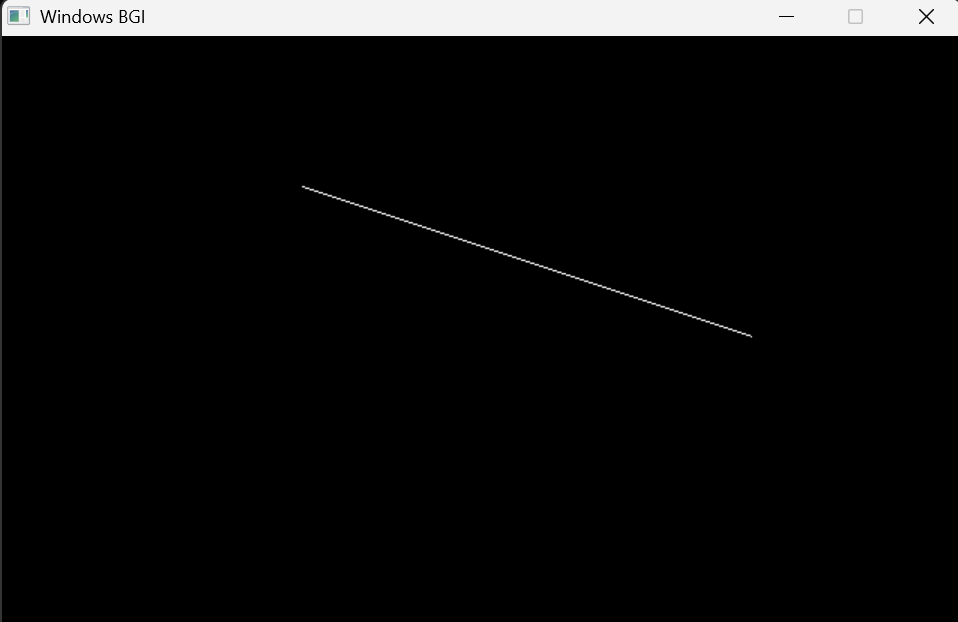
For positive slope & |m|>1



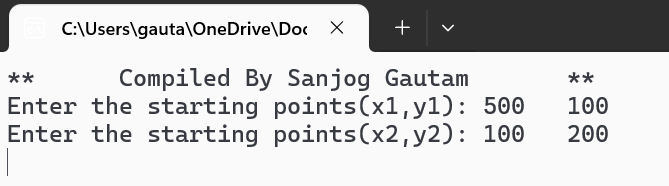


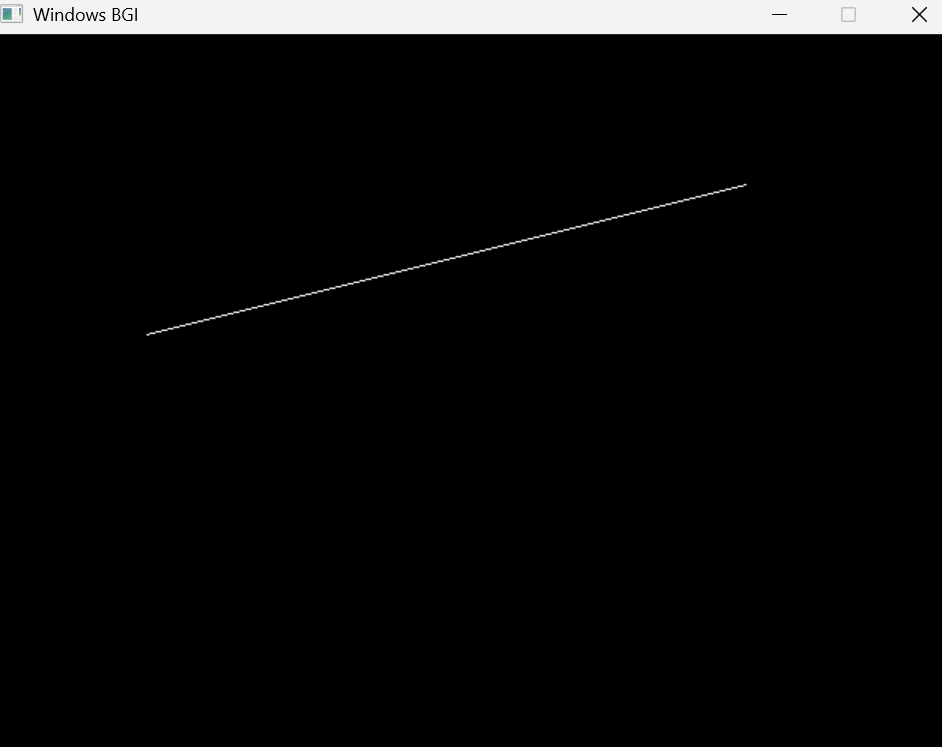
For positive slope & |m|<1



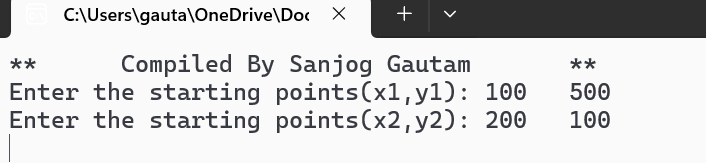


For negative slope & |m|<1





For negative slope & |m|>1





**Lab 2:** WAP in C to implement BLA algorithm for:

1. |m|>1 positive slope
2. |m|>1 negative slope

Source Code:

#include<stdio.h>

#include<graphics.h>

#include<math.h>

#include<conio.h>

int main()

{

printf("\*\*\tCompiled by Sanjog Gautam\t\*\*\n");

int gm,gd=DETECT;

int x1,y1,x2,y2,i,j,Pk;

float m,x,y;

printf("Enter the initial coordinate(x1,y1): ");

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

printf("Enter the final coordinate(x2,y2): ");

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

initgraph(&gd,&gm," ");

int dx=x2-x1;

int dy=y2-y1;

m=float(dy)/dx;

dx=fabs(dx);

dy=fabs(dy);

if(fabs(m)>1) //for slope:|m|>1;

{

float P0=2\*dx-dy;

x=x1;

y=y1;

if(m>=0)// for positive slope & |m|>1

{

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

{

if(P0<0)

{

x=x;

y=y+1;

putpixel(x,y,WHITE);

delay(10);

P0=P0+2\*dx;

}

else

{

x=x+1;

y=y+1;

putpixel(x,y,WHITE);

delay(10);

P0=P0+2\*dx-2\*dy;

}

}}

else

{

for(i=0;i<=dy;i++)// for negative slope & |m|>1

{

if(P0<0)

{

x=x;

y=y-1;

putpixel(x,y,WHITE);

delay(10);

P0=P0+2\*dx;

}

else

{

x=x+1;

y=y-1;

putpixel(x,y,WHITE);

delay(10);

P0=P0+2\*dx-2\*dy;

}

}

}

}

else //for slope:|m|<1

{

float P0=2\*dy-dx;

x=x1;

y=y1;

if(m>=0)// for positive slope & |m|<1

{

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

{

if(P0<0)

{

x=x+1;

y=y;

putpixel(x,y,WHITE);

delay(10);

P0=P0+2\*dy;

}

else

{

x=x+1;

y=y+1;

putpixel(x,y,WHITE);

delay(10);

P0=P0+2\*dy-2\*dx;

}

}}

else

{

for(i=0;i<=dx;i++)// for negative slope & |m|<1

{

if(P0<0)

{

x=x+1;

y=y-1;

putpixel(x,y,WHITE);

delay(10);

P0=P0+2\*dx;

}

else

{

x=x+1;

y=y;

putpixel(x,y,WHITE);

delay(10);

P0=P0+2\*dx-2\*dy;

}

}

}

}

getch();

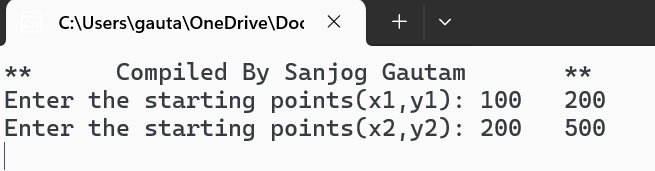
closegraph();

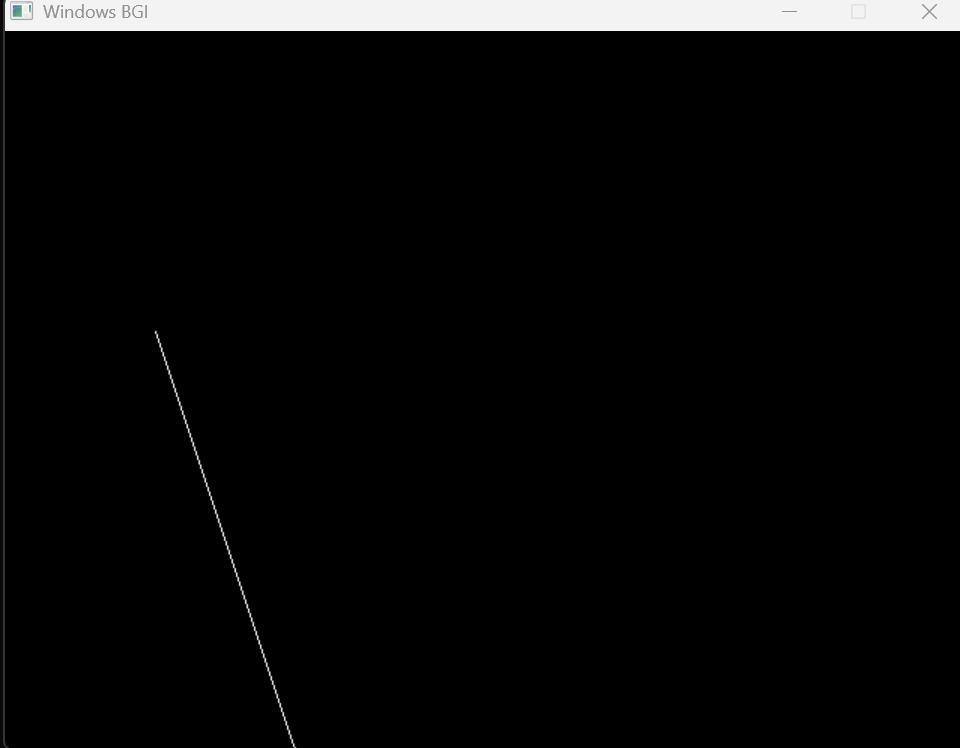
return 0;

}

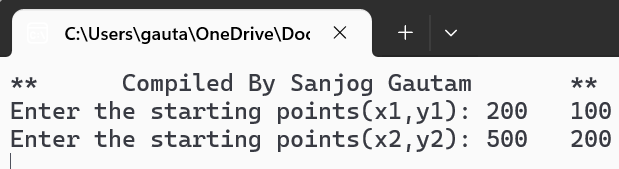
Output:

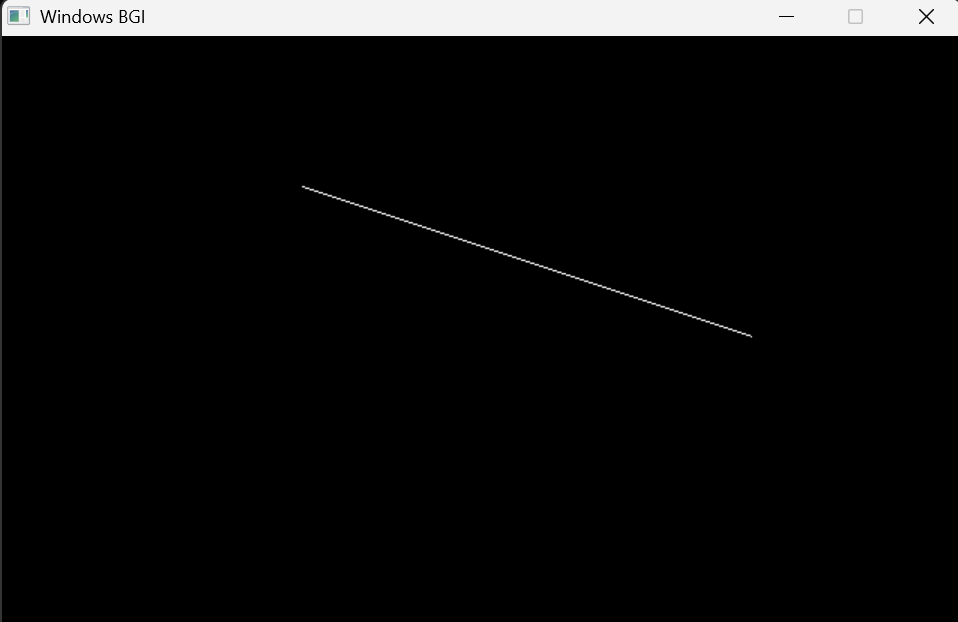
For positive slope & |m|>1



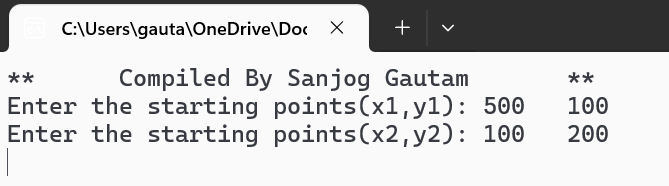


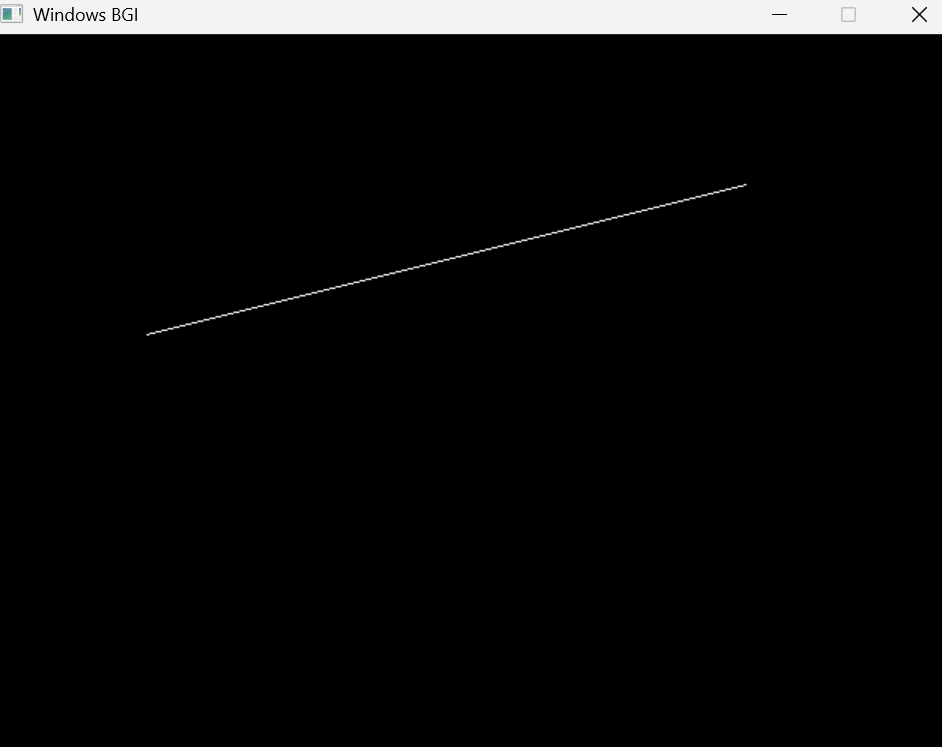
For positive slope & |m|<1





For negative slope & |m|<1





For negative slope & |m|>1

