Name:Swaraj Sambhaji Gore
Roll No.:25

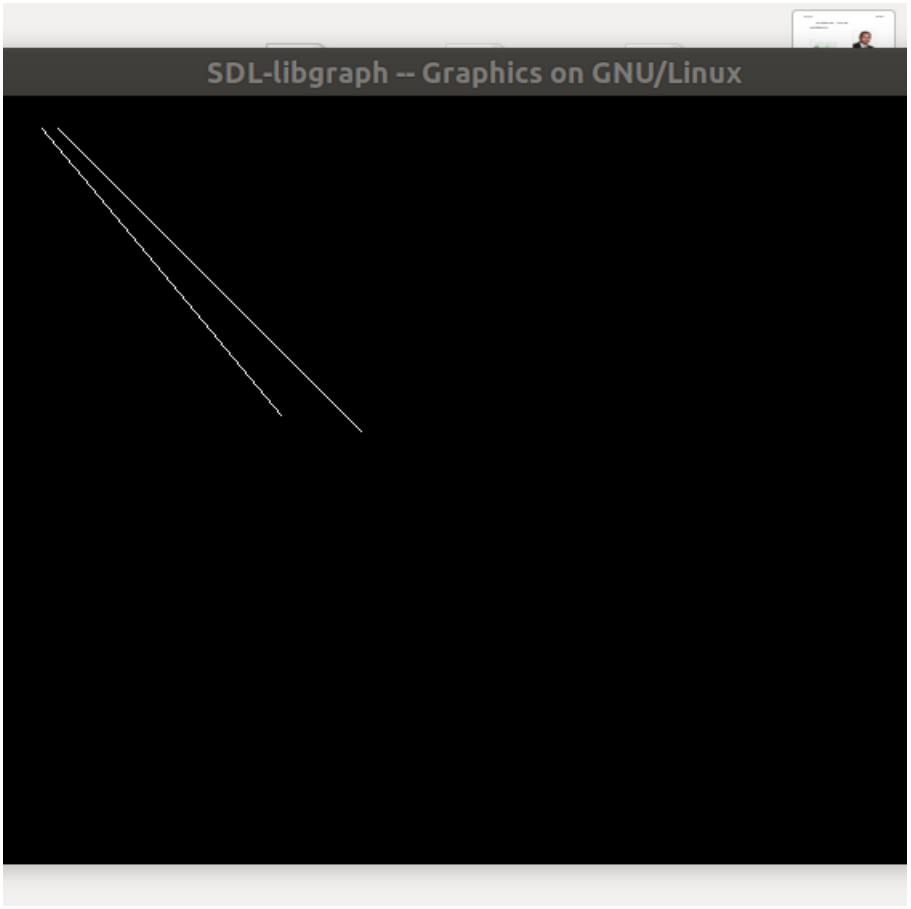
```
/*
1.Write C++/Java program to draw line using DDA and Bresenham‘s
algorithm. Inherit pixel
class and Use function overloading.
*/
#include<iostream>
#include<math.h>
#include<graphics.h>
using namespace std;
class pixel
{
public:float x1,y1,x2,y2,dx,dy,step;
float xinc,yinc,x,y;
int bx1,by1,bx2,by2,dx1,dy1,p0,p1,t,t1,t2,steps;
public: void drawline(int,int,int,int,int);
void drawline(float,float,float,float);
};
class point:public pixel
{
pixel p;
public:
void getdata1()
{
cout<<"enter cordinates of x1 y1 x2 y2"<<endl;
cin>>p.x1>>p.y1>>p.x2>>p.y2;
p.drawline(p.x1,p.y1,p.x2,p.y2);
}
void getdata2()
{
cout<<"\n cordinates"<<endl;
cin>>p.bx1>>p.by1>>p.bx2>>p.by2;
p.drawline(p.bx1,p.by1,p.bx2,p.by2,p.dx1);
}
};
void pixel::dataline(int bx1,int by1,int bx2,int by2,int dx1)
{
dx1=bx2-bx1;
dy1=by2-by1;
t=2*dx1;
t1=2*dy1;
t2=t1-dx1;
p0=t2;
putpixel(bx1,by1,WHITE);
for(int i=0;i<dx1-1;i++)
{
bx1=bx1+1;
if(p0<0)
{
p1=p0+t1;
}
```

```
else
{
by1=by1+1;
p1=p0+t1-t;
}
putpixel(bx1,by1,WHITE);
}
}
void pixel::drawline(float x1,float y1,float x2,float y2)
{
dx=x2-x1;
dy=y2-y1;
if(dx>dy)
{
step=dx;
}
else
{
step=dy;
}
xinc=dx/step;
yinc=dy/step;
x=x1;
y=y1;
for(int i=0;i<step;i++)
{
putpixel(round(x),round(y),WHITE);
x=x+xinc;
y=y+yinc;
}
}
int main()
{
int gd=DETECT,gm;
initgraph(&gd,&gm,NULL);
point pt;
int ch;
char ans;
do
{
cout<<"\n MENU:\n 1. DDA \n 2.Bresenham "<<endl;
cin>>ch;
switch(ch)
{
case 1:cout<<"DDA"<<endl;
       pt.getdata1();
       break;
case 2: cout<<"Bresenham"<<endl;
       pt.getdata2();
       break;
}
cout<<"\nDo you want to continue?(y/n)"<<endl;
```

```
cin>>ans;
}while(ans=='y');
closegraph();
return 0;
}
```

Output:

```
 MENU:
 1. DDA
 2.Bresenham
1
DDA
enter cordinates of x1 y1 x2 y2
50
20
200
200

Do you want to continue?(y/n)
y

 MENU:
 1. DDA
 2.Bresenham

Do you want to continue?(y/n)
y

 MENU:
 1. DDA
 2.Bresenham
2
Bresenham

 cordinates
60
20
250
250

Do you want to continue?(y/n)
```


SDL-libgraph -- Graphics on GNU/Linux