```
#include<iostream>
#include<stdio.h>
#include<math.h>
#include<graphics.h>
#define ROUND(a)((int)(a+0.5))
using namespace std;

class pixel
{
protected:int x1,x2,y1,y2;
void accept();
};
void pixel::accept()
{
cout<<"Enter the starting coordinates"<<endl;
cin>>x1;
cin>>y1;
cout<<"Enter the destination coordinates"<<endl;
cin>>x2;
cin>>y2;
}

class draw:public pixel
{
private:
void dotted();
void dashed();
void dash_dott();
void solid();
void thick_line(int xa,int ya,int xb,int yb);
public:
void normal_call();
};
void draw::dotted()
{
int gd=DETECT,gm;
initgraph(&gd,&gm,NULL);
int dx,dy,steps;
if(x2>x1)
{
dx=x2-x1;
}
else
{
dx=x1-x2;
}
if(y2>y1)
{
dy=y2-y1;
}
else
{
dy=y1-y2;
}
if(dx>=dy)
steps=dx;
else
steps=dy;
dx=dx/steps;
int sx;
if(dx>=0)
sx=1;
else
sx=-1;
dy=dy/steps;
int sy;
```

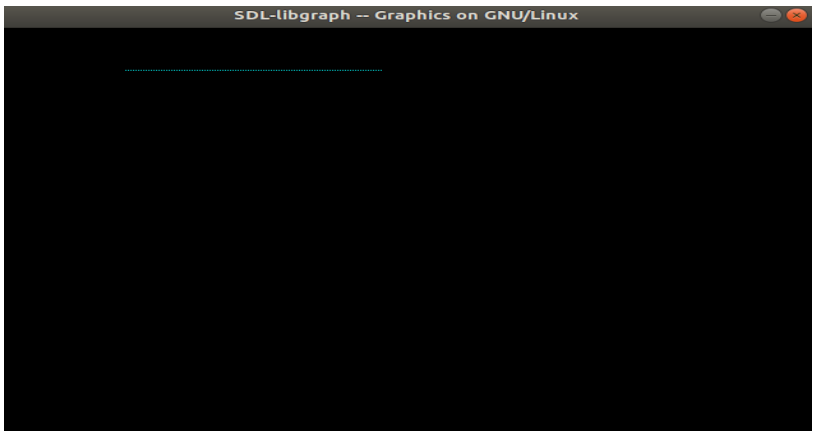
```
if(dy>=0)
sy=1;
else
sy=-1;
float x=x1+0.5*sx;
float y=y1+0.5*sy;
int i=0;
while(i<=steps)
{
if(i%2==0)
{
putpixel(int(x),int(y),3);
}
x=x+dx;
y=y+dy;
i++;
}
getch();
closegraph();
}
void draw::dashed()
{
int gd=DETECT,gm;
initgraph(&gd,&gm,NULL);
int dx,dy,steps;
if(x2>x1)
{
dx=x2-x1;
}
else
{
dx=x1-x2;
}
if(y2>y1)
{
dy=y2-y1;
}
else
{
dy=y1-y2;
}
if(dx>=dy)
steps=dx;
else
steps=dy;
dx=dx/steps;
int sx;
if(dx>=0)
sx=1;
else
sx=-1;
dy=dy/steps;
int sy;
if(dy>=0)
sy=1;
else
sy=-1;
float x=x1+0.5*sx;
float y=y1+0.5*sy;
int i=0;
while(i<=steps)
{
if(i%9<2)
{
}
 else if(i%9<6)
```

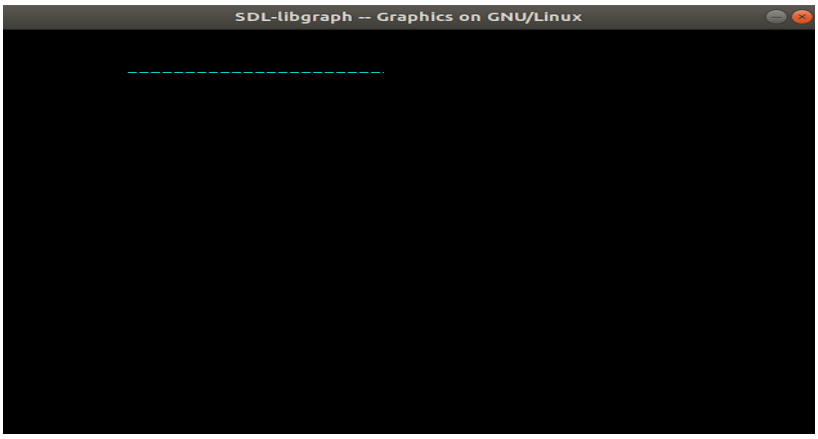
```
{
putpixel(int(x),int(y),3);
}
else
putpixel(int(x),int(y),3);
x=x+dx;
y=y+dy;
i++;
}
getch();
closegraph();
}
void draw::dash_dott()
{
int gd=DETECT,gm;
initgraph(&gd,&gm,NULL);
int dx,dy,steps;
if(x2>x1)
{
dx=x2-x1;
}
else
{
dx=x1-x2;
}
if(y2>y1)
{
dy=y2-y1;
}
else
{
dy=y1-y2;
}
if(dx>=dy)
steps=dx;
else
steps=dy;
dx=dx/steps;
int sx;
if(dx>=0)
sx=1;
else
sx=-1;
dy=dy/steps;
int sy;
if(dy>=0)
sy=1;
else
sy=-1;
float x=x1+0.5*sx;
float y=y1+0.5*sy;
int i=0;
while(i<=steps)
{
if(i%9<2)
{
}
 else if(i%9<6)
{
putpixel(int(x),int(y),3);
}
else if(i%9==7)
{
}
else
putpixel(int(x),int(y),3);
```

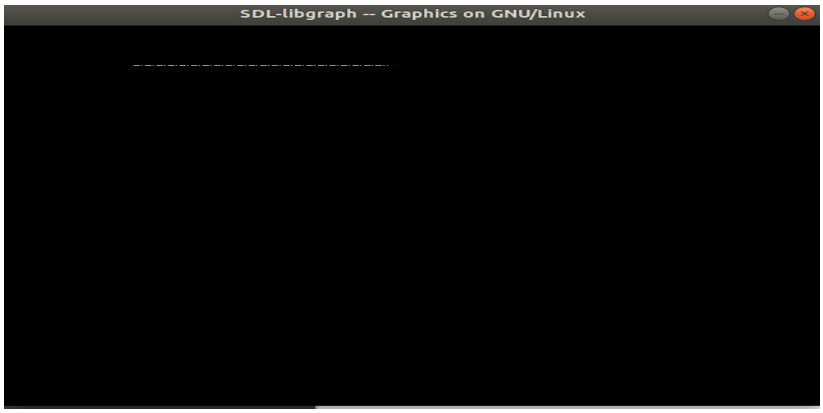
```
x=x+dx;
y=y+dy;
i++;
}
getch();
closegraph();
}
void draw::solid()
{
int gd=DETECT,gm;
initgraph(&gd,&gm,NULL);
int dx,dy,steps;
if(x2>x1)
{
dx=x2-x1;
}
else
{
dx=x1-x2;
}
if(y2>y1)
{
dy=y2-y1;
}
else
{
dy=y1-y2;
}
if(dx>=dy)
steps=dx;
else
steps=dy;
dx=dx/steps;
int sx;
if(dx>=0)
sx=1;
else
sx=-1;
dy=dy/steps;
int sy;
if(dy>=0)
sy=1;
else
sy=-1;
float x=x1+0.5*sx;
float y=y1+0.5*sy;
int i=0;
while(i<=steps)
{
putpixel(int(x),int(y),3);
x=x+dx;
y=y+dy;
i++;
}
getch();
closegraph();
}


void draw::thick_line(int xa,int ya,int xb,int yb)
{
int gd=DETECT,gm;
initgraph(&gd,&gm,NULL);
int dx,dy,steps;
dx=xb-xa;
dy=yb-ya;
```

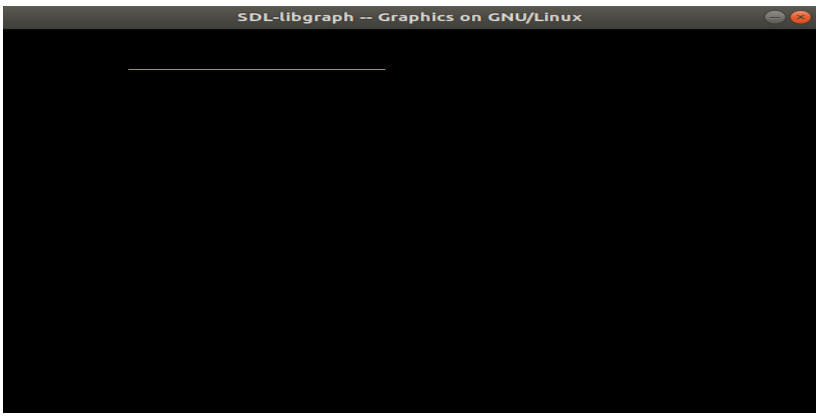
```
if(abs(dx)>=abs(dy))
steps=abs(dx);
else
steps=abs(dy);
float x=xa,y=ya,xinc,yinc;
xinc=dx/float(steps);
yinc=dy/float(steps);
putpixel(ROUND(x),ROUND(y),3);
for(int i=0;i<steps;i++)
{
x=x+xinc;
y=y+yinc;
putpixel(ROUND(x),ROUND(y),3);
}
getch();
closegraph();
}




void draw::normal_call()
{
accept();
int ch;
do
{
cout<<"1.Dotted   2.Dashed    3.Dash-dott     4.Solid(Regular)
5.Thick_line     6.Exit"<<endl;
cin>>ch;
switch(ch)
{
case 1:dotted();
restorecrtmode();
break;
case 2:dashed();
restorecrtmode();
break;
case 3:dash_dott();
restorecrtmode();
break;
case 4:solid();
restorecrtmode();
break;
case 5:
{
int thick;
cout<<"Enter the thickness"<<endl;
cin>>thick;
int dy=y2-y1;
int dx=x2-x1;
int wy,wx;
if((dy/dx)<1)
      {
         wy=(thick-1)*(sqrt(pow(x2-x1,2))+sqrt(pow(y2-y1,2)))/2*(x2-x1);
      for(int j=0;j<wy;j++)
         {
      thick_line(x1,y1-j,x2,y2-j);
         delay(2000);
      thick_line(x1,y1+j,x2,y2+j);
      }
      }
      else
         {
      wx=(thick-1)*(sqrt(pow(x2-x1,2))+sqrt(pow(y2-y1,2)))/2*(x2-x1);
```

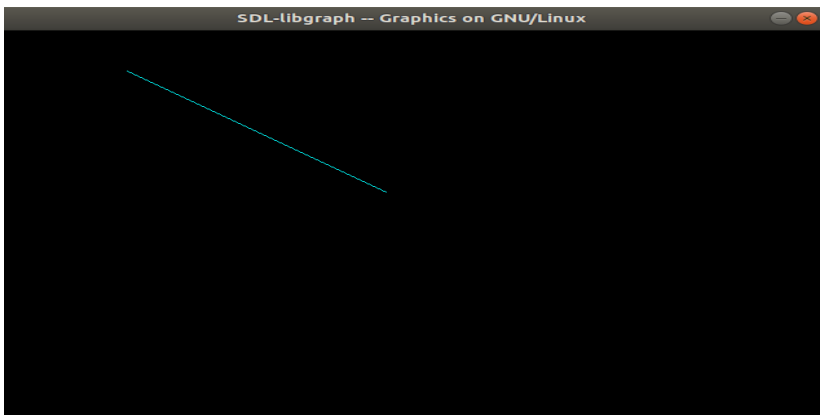
```
        for(int j=0;j<wx;j++)
      {
      thick_line(x1-j,y1,x2-j,y2);
      delay(2000);
      thick_line(x1+j,y1,x2+j,y2);
      }
}

}
break;
}
}while(ch!=6);
}
int main()
{
draw d;
d.normal_call();
return 0;
}
Enter the starting coordinates
100
50
Enter the destination coordinates
300
200
1.Dotted   2.Dashed    3.Dash-dott     4.Solid(Regular)    5.Thick_line
6.Exit
1
1.Dotted   2.Dashed    3.Dash-dott     4.Solid(Regular)    5.Thick_line
6.Exit
3
1.Dotted   2.Dashed    3.Dash-dott     4.Solid(Regular)    5.Thick_line
6.Exit
4
Enter the starting coordinates
100
50
Enter the destination coordinates
300
200
1.Dotted   2.Dashed    3.Dash-dott     4.Solid(Regular)    5.Thick_line
6.Exit
5
Enter the thickness5
```




SDL-libgraph -- Graphics on GNU/Linux


SDL-libgraph -- Graphics on GNU/Linux


SDL-libgraph -- Graphics on GNU/Linux


SDL-libgraph -- Graphics on GNU/Linux