CG Experiment No.

Aim: Implementation of Bresenham's Line Drawing Algorith

Theory: The Bresenham's Line Algorithm is an algorithm which determines which point is an n-dimensional roaster should be plotted in order to form a close approximation to a straight line between two points. The given endpoints of the line are the pixels at (x_0, y_0) and (x_1, y_1), where the first co-ordinate of the pair is the column and the second is the row.

Example: Consider the line from (5,5) to (13,9) use the Bresenham's algorithm to rasterize the line.

Evaluating step 1 through y in the Bresenham's algorithm we have,

$$\Delta x = |13-5| = 8$$

$$\Delta y = |9-5| = 4.$$

$x=5, y=5.$

$$e = 2^{*}\Delta y - \Delta x = 2^{*}4-8=0.$$

Tabulating the results of each iteration in step 5 through 10.

i	Plot	x	y	e
		5	5	0
	(5,5)	6	6	-8
1	(6,6)	7	6	0
2	(7,6)	8	7	-8
3	(8,7)	9	7	0
4	(9,7)	10	8	-8
5	(10,8)	4	8	0
6	(11,8)	12	9	-8
7	(12,9)	13	9	0
8	(13,9)	14	10	-8

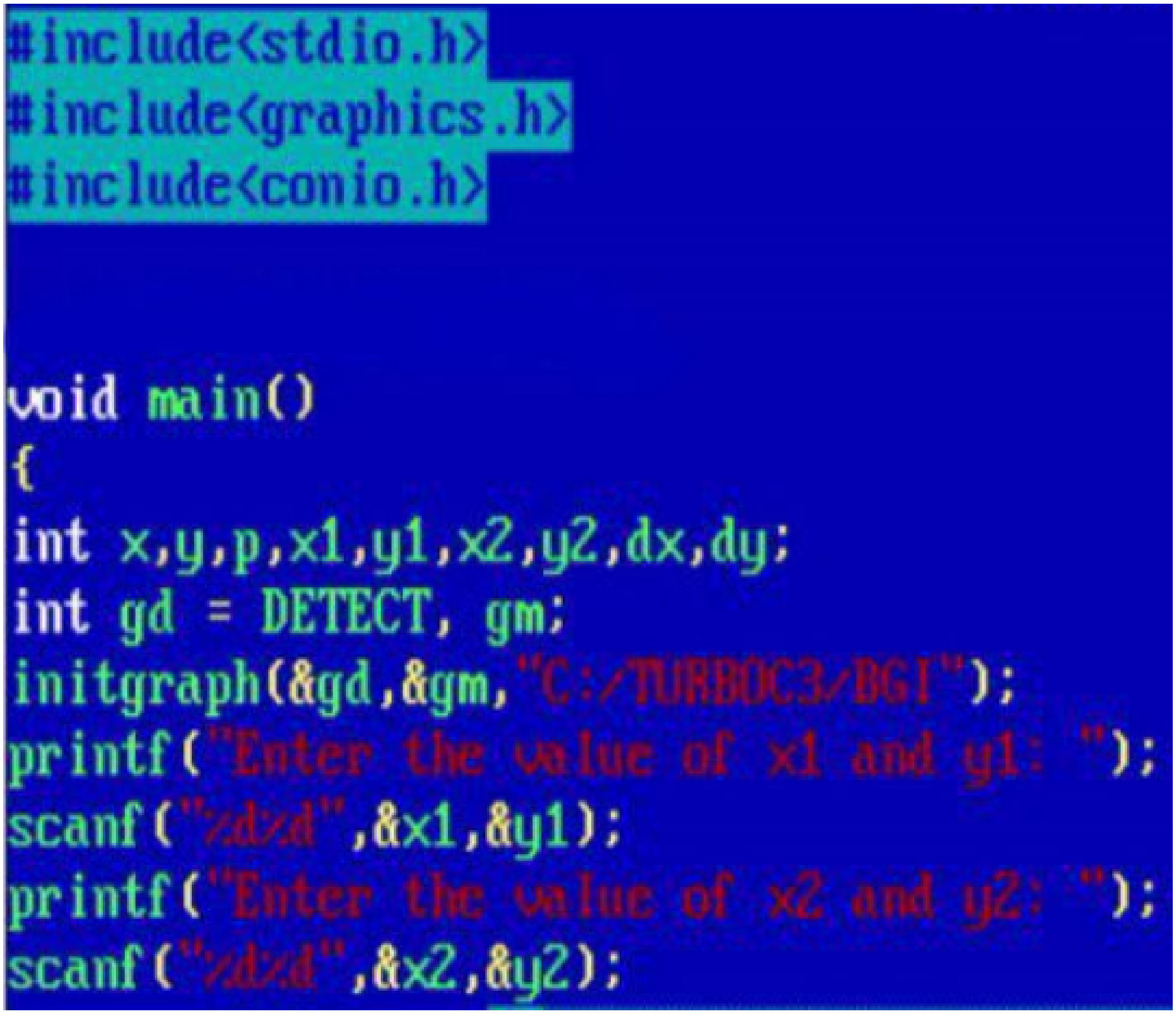
Algorithm for $|m = \Delta y/\Delta x| < 1$

```
1. Read the line endpoints $(x_1, y_1)$ and $(x_2, y_2)$ such they are unequal
2. $\Delta x = |x_2 - x_1|$   and   $\Delta y = |y_2 - y_1|$.
3. (Initialising start point)   $x = x_1$ and $y = y_1$
                                Plot $(x, y)$
4. $e = 2 * \Delta y - \Delta x$ (Initialise value of decision variable or
        error to compensate for non-zero interropts)
5. $i = 1$ (Initialise counter)
6. While $(e \geq 0)$ { $y = y + 1$;
                        $e = e - 2 * \Delta x$; }
      $x = x + 1$;   $e = e + 2 * \Delta y$;
7. Plot $(x, y)$
8. $i = i + 1$
9. If $(i \leq \Delta x)$ then goto Step 6.
10. Stop.
```

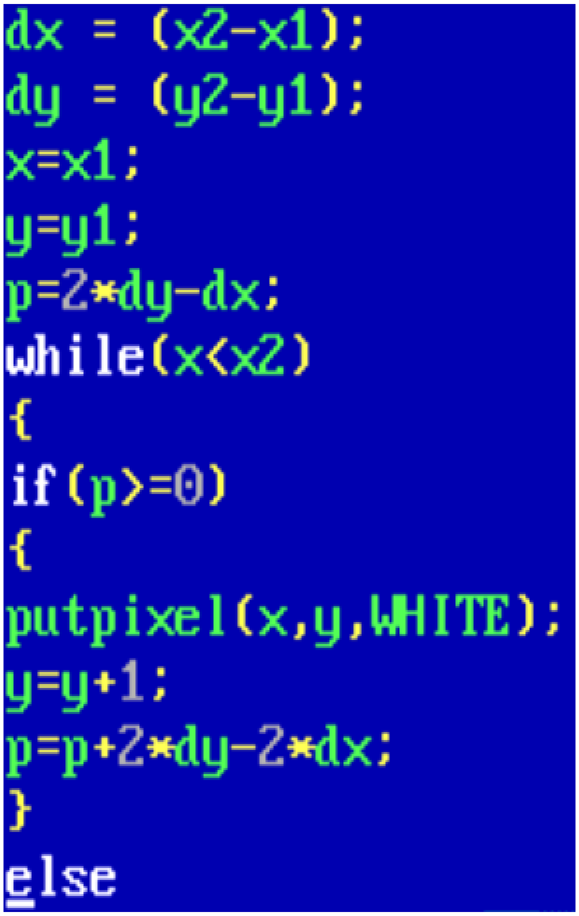
Topic: Computer Graphics LAB 3 – Implement Bresenham’s line drawing algorithm(Thick,Dash,Dotted).

A) Write a program in C to draw a thick line using Bresenham’s algorithm.

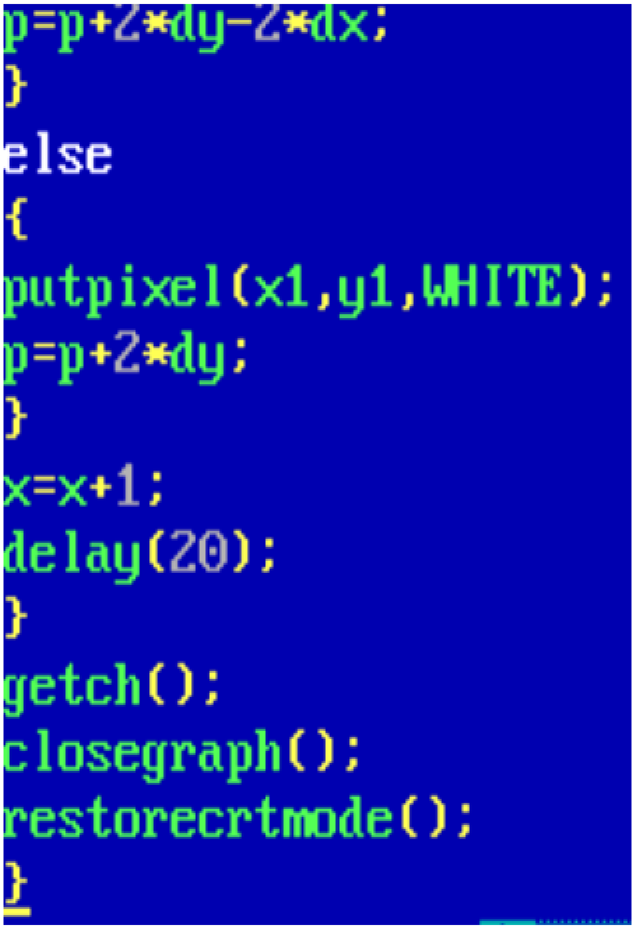
Program :



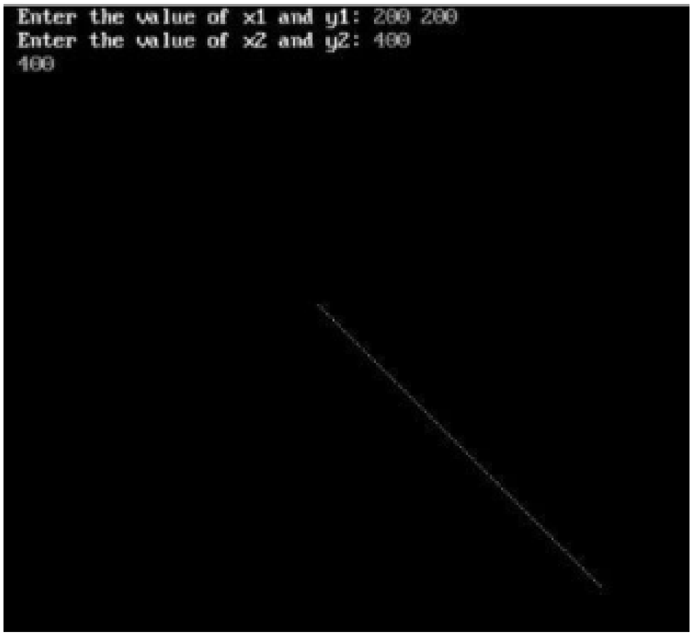
```
#include<stdio.h>
#include<graphics.h>
#include<conio.h>


void main()
{
int x,y,p,x1,y1,x2,y2,dx,dy;
int gd = DETECT, gm;
initgraph(&gd,&gm,"C:/TURBOC3/BGI");
printf("Enter the value of x1 and y1: ");
scanf("%d%d",&x1,&y1);
printf("Enter the value of x2 and y2: ");
scanf("%d%d",&x2,&y2);
```



```
dx = (x2-x1);
dy = (y2-y1);
x=x1;
y=y1;
p=2*dy-dx;
while(x<x2)
{
if(p>=0)
{
putpixel(x,y,WHITE);
y=y+1;
p=p+2*dy-2*dx;
}
else
```



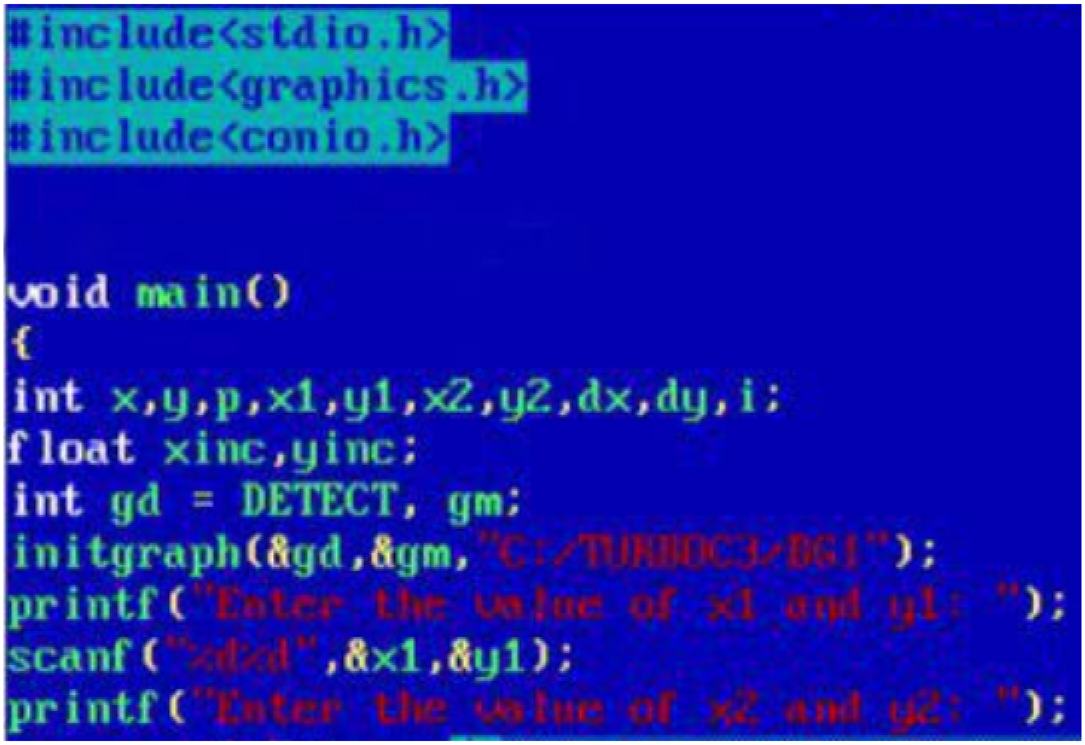
```
p=p+2*dy-2*dx;
}
else
{
putpixel(x1,y1,WHITE);
p=p+2*dy;
}
x=x+1;
delay(20);
}
getch();
closegraph();
restorecrtmode();
}
```



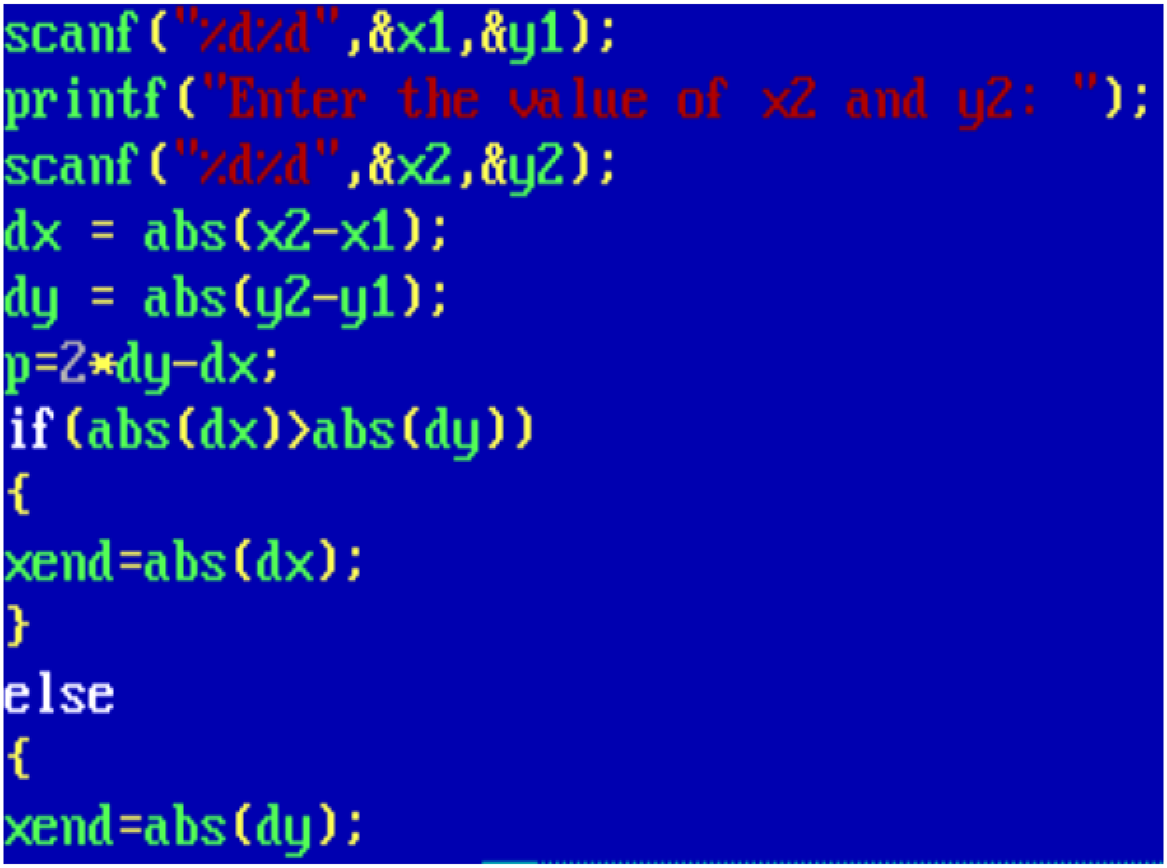
```
Enter the value of x1 and y1: 200 200
Enter the value of x2 and y2: 400
400
```

B)Write a program in C to draw a dotted line using Bresenham’s algorithm.

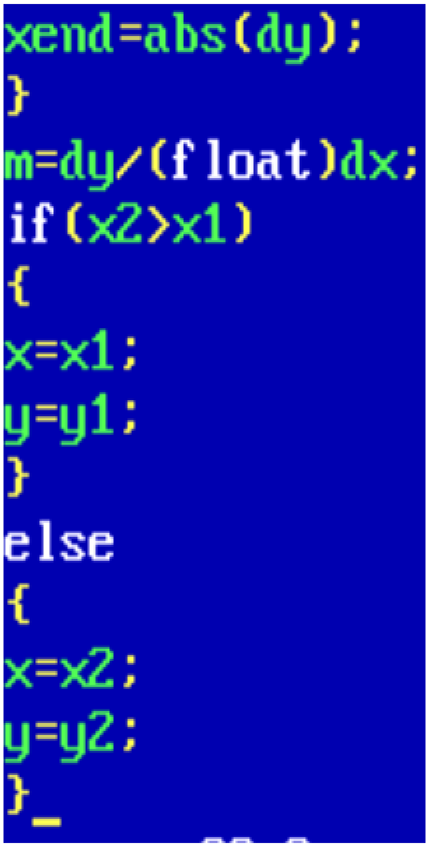
Program :



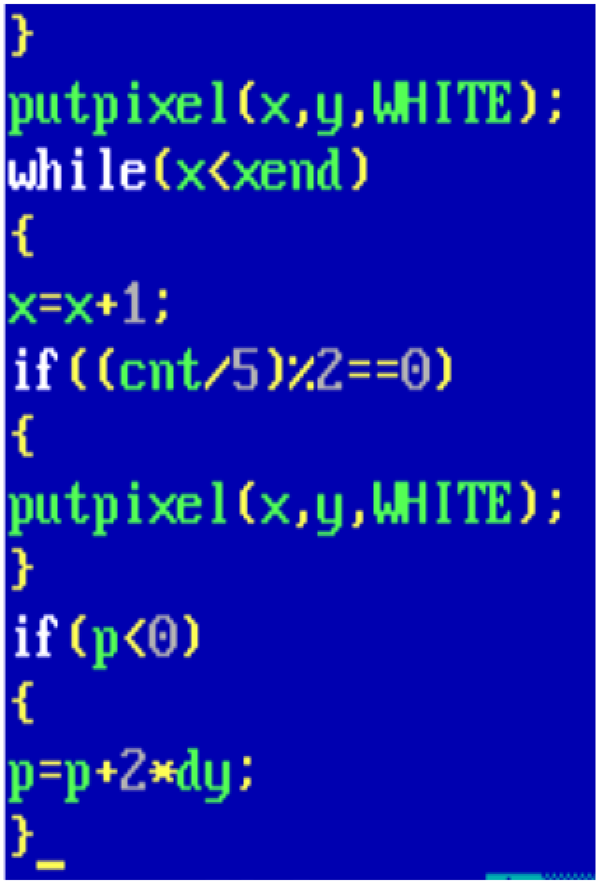
```
#include<stdio.h>
#include<graphics.h>
#include<conio.h>


void main()
{
int x,y,p,x1,y1,x2,y2,dx,dy,i;
float xinc,yinc;
int gd = DETECT, gm;
initgraph(&gd,&gm,"C:/TURBOC3/BGI");
printf("Enter the value of x1 and y1: ");
scanf("%d%d",&x1,&y1);
printf("Enter the value of x2 and y2: ");
```



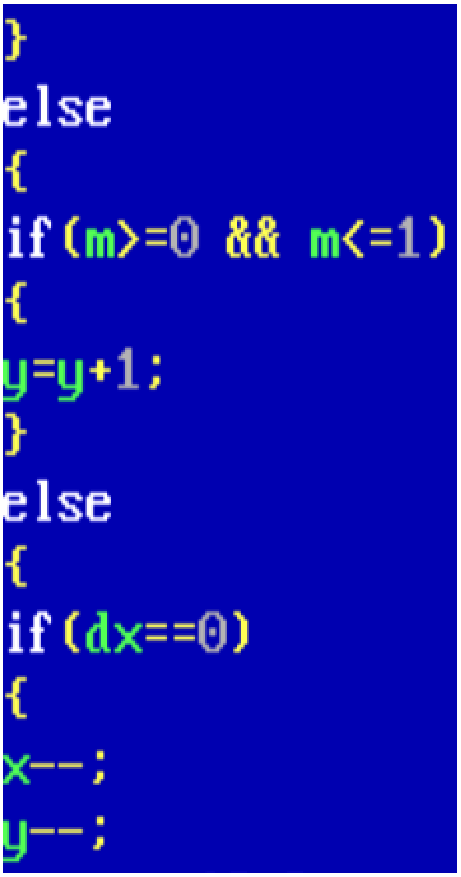
```
scanf("%d%d",&x1,&y1);
printf("Enter the value of x2 and y2: ");
scanf("%d%d",&x2,&y2);
dx = abs(x2-x1);
dy = abs(y2-y1);
p=2*dy-dx;
if(abs(dx)>abs(dy))
{
xend=abs(dx);
}
else
{
xend=abs(dy);
```



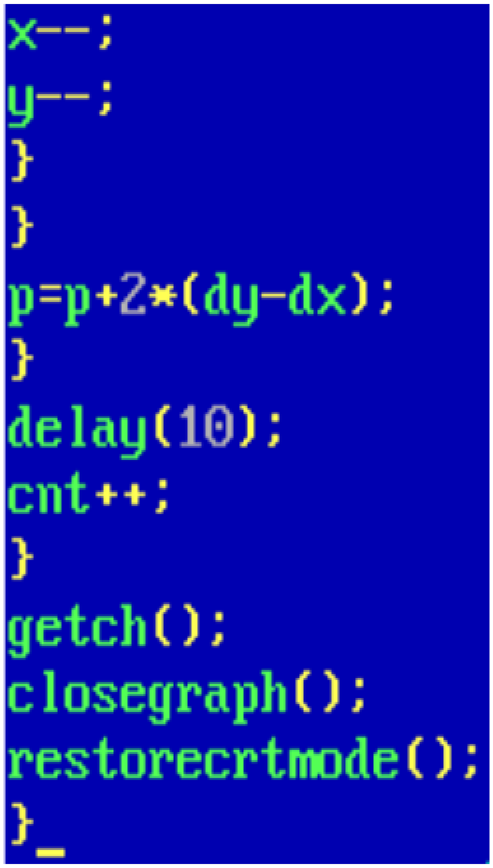
```
xend=abs(dy);
}
m=dy/(float)dx;
if(x2>x1)
{
x=x1;
y=y1;
}
else
{
x=x2;
y=y2;
}
```



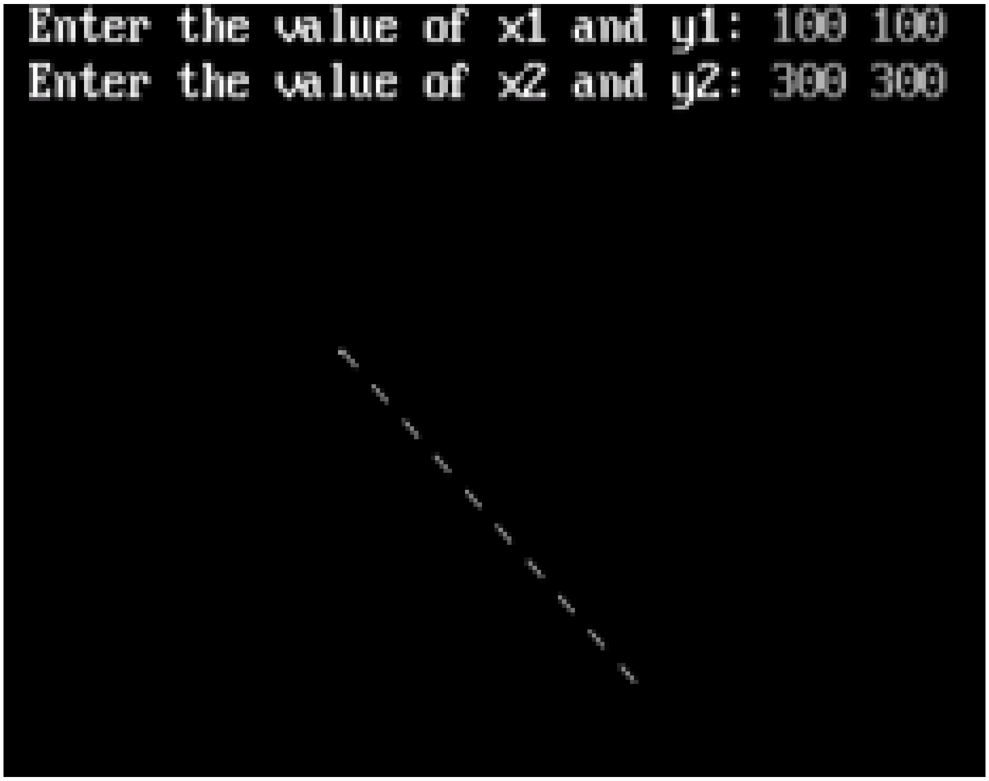
```
}
putpixel(x,y,WHITE);
while(x<xend)
{
x=x+1;
if((cnt/5)%2==0)
{
putpixel(x,y,WHITE);
}
if(p<0)
{
p=p+2*dy;
}
```



```
}
else
{
if(m>=0 && m<=1)
{
y=y+1;
}
else
{
if(dx==0)
{
x--;
y--;
```



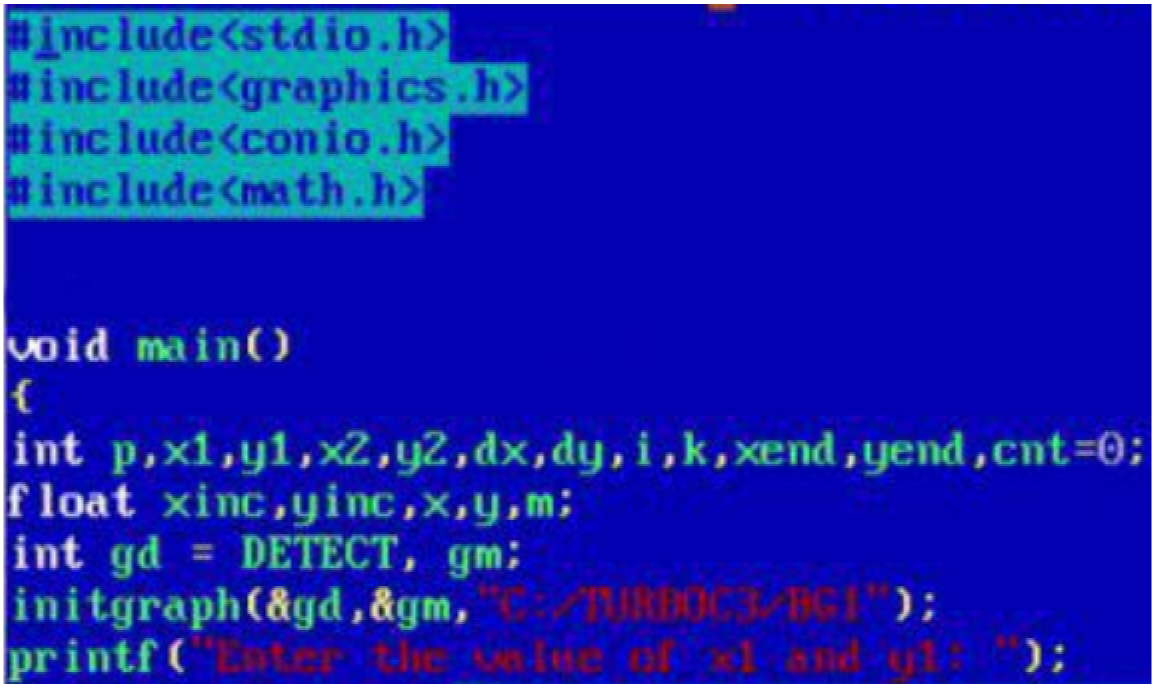
```
x--;
y--;
}
}
p=p+2*(dy-dx);
}
delay(10);
cnt++;
}
getch();
closegraph();
restorecrtmode();
}_
```



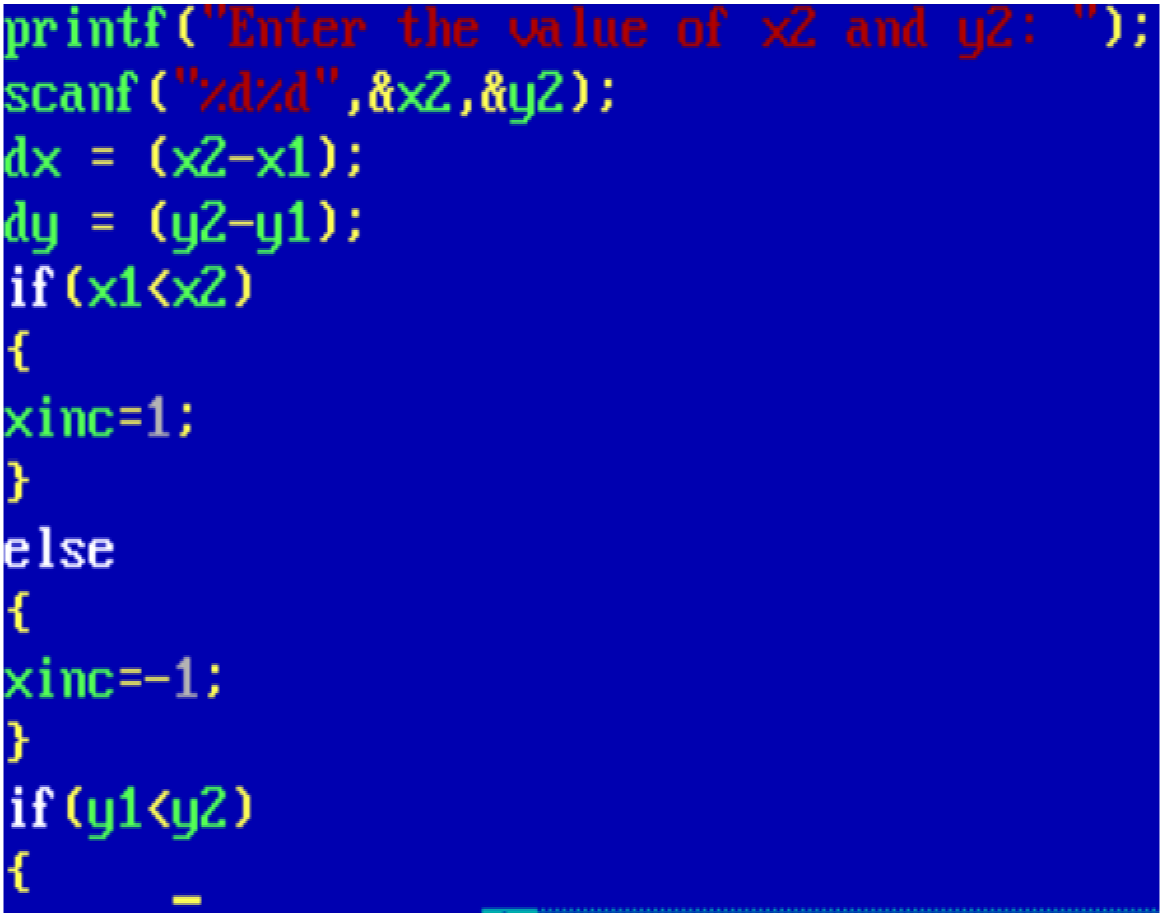
```
Enter the value of x1 and y1: 100 100
Enter the value of x2 and y2: 300 300
```

C) Write a program in C to draw a dash line using Bresenham's algorithm.

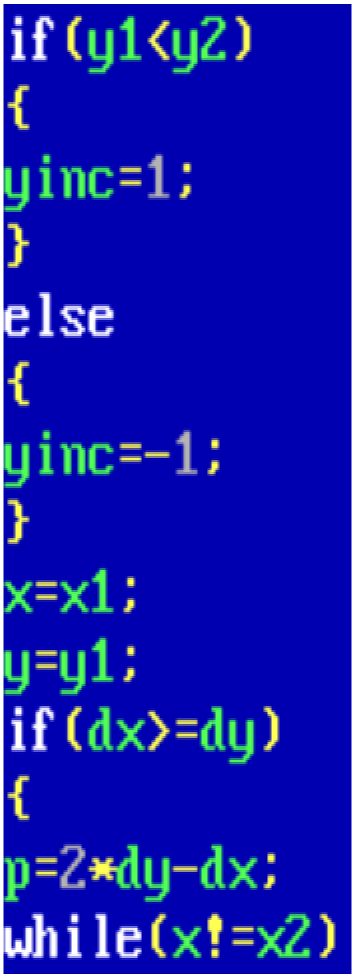
Program :



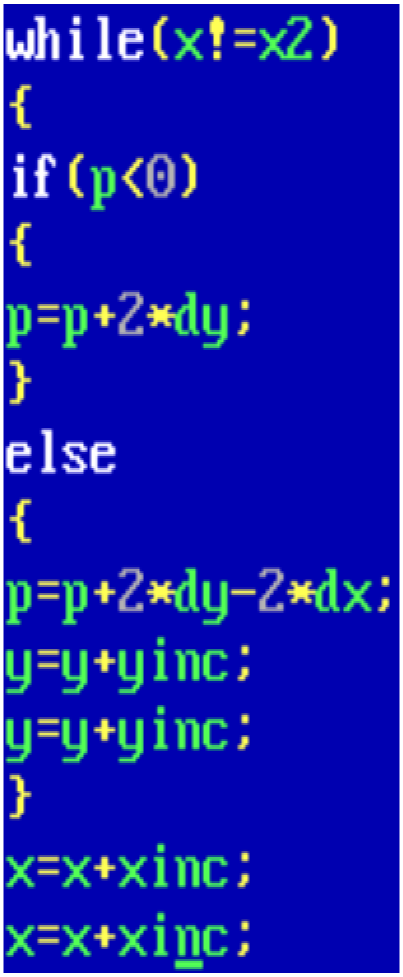
```
#include<stdio.h>
#include<graphics.h>
#include<conio.h>
#include<math.h>


void main()
{
int p,x1,y1,x2,y2,dx,dy,i,k,xend,yend,cnt=0;
float xinc,yinc,x,y,m;
int gd = DETECT, gm;
initgraph(&gd,&gm,"C:/TURBOC3/BGI");
printf("Enter the value of x1 and y1: ");
```



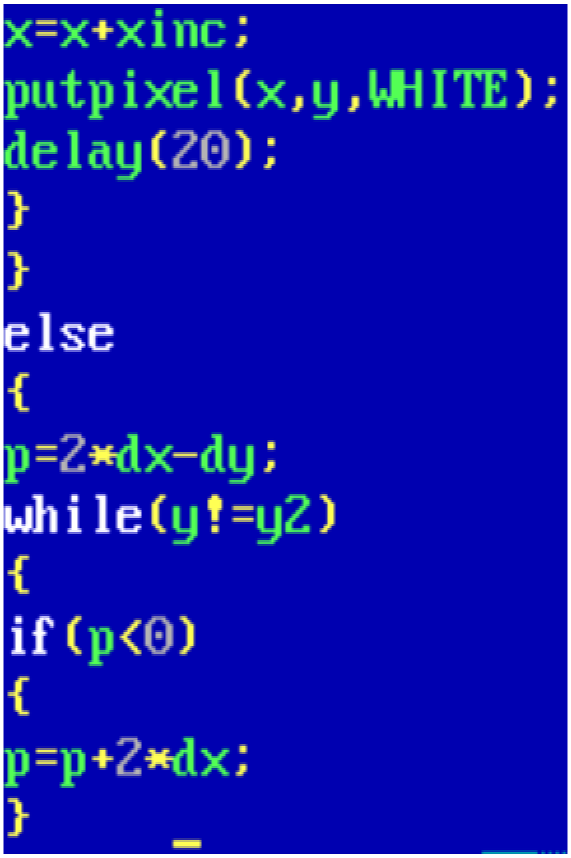
```
printf("Enter the value of x2 and y2: ");
scanf("%d%d",&x2,&y2);
dx = (x2-x1);
dy = (y2-y1);
if(x1<x2)
{
xinc=1;
}
else
{
xinc=-1;
}
if(y1<y2)
{
```



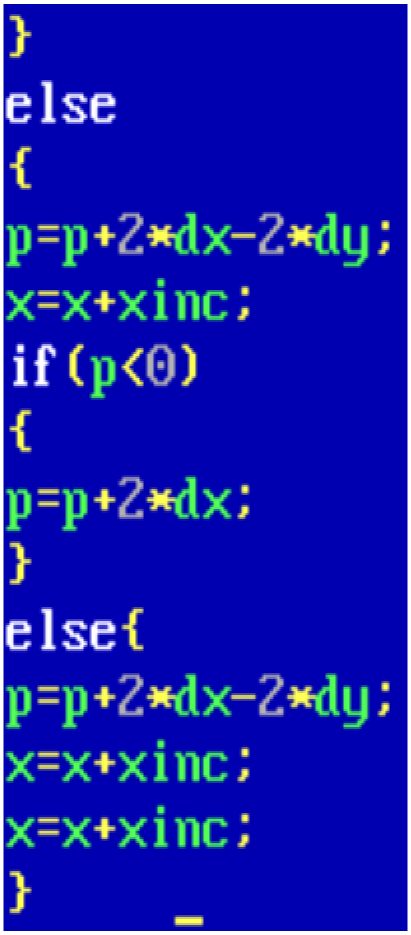
```
if(y1<y2)
{
yinc=1;
}
else
{
yinc=-1;
}
x=x1;
y=y1;
if(dx>=dy)
{
p=2*dy-dx;
while(x!=x2)
```



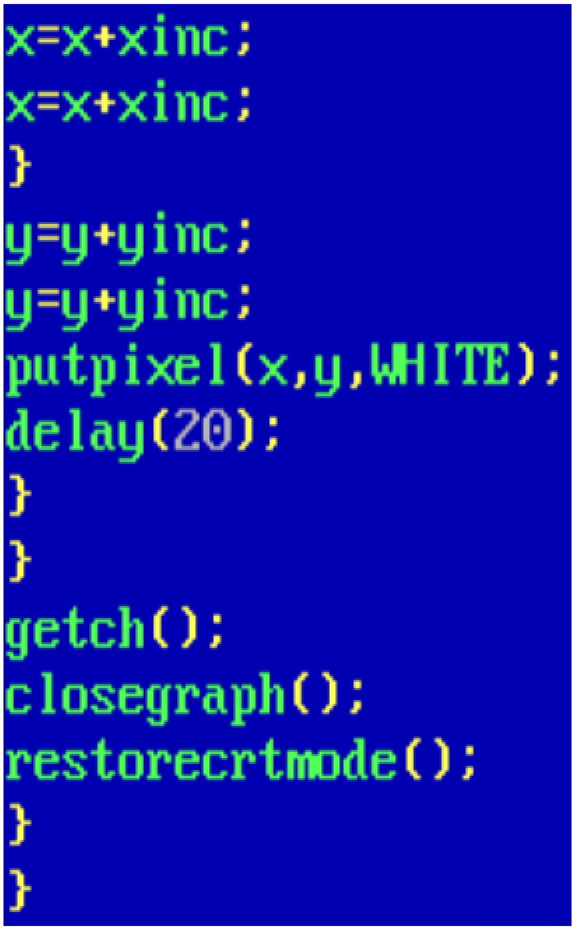
```
while(x!=x2)
{
if(p<0)
{
p=p+2*dy;
}
else
{
p=p+2*dy-2*dx;
y=y+yinc;
y=y+yinc;
}
x=x+xinc;
x=x+xinc;
```



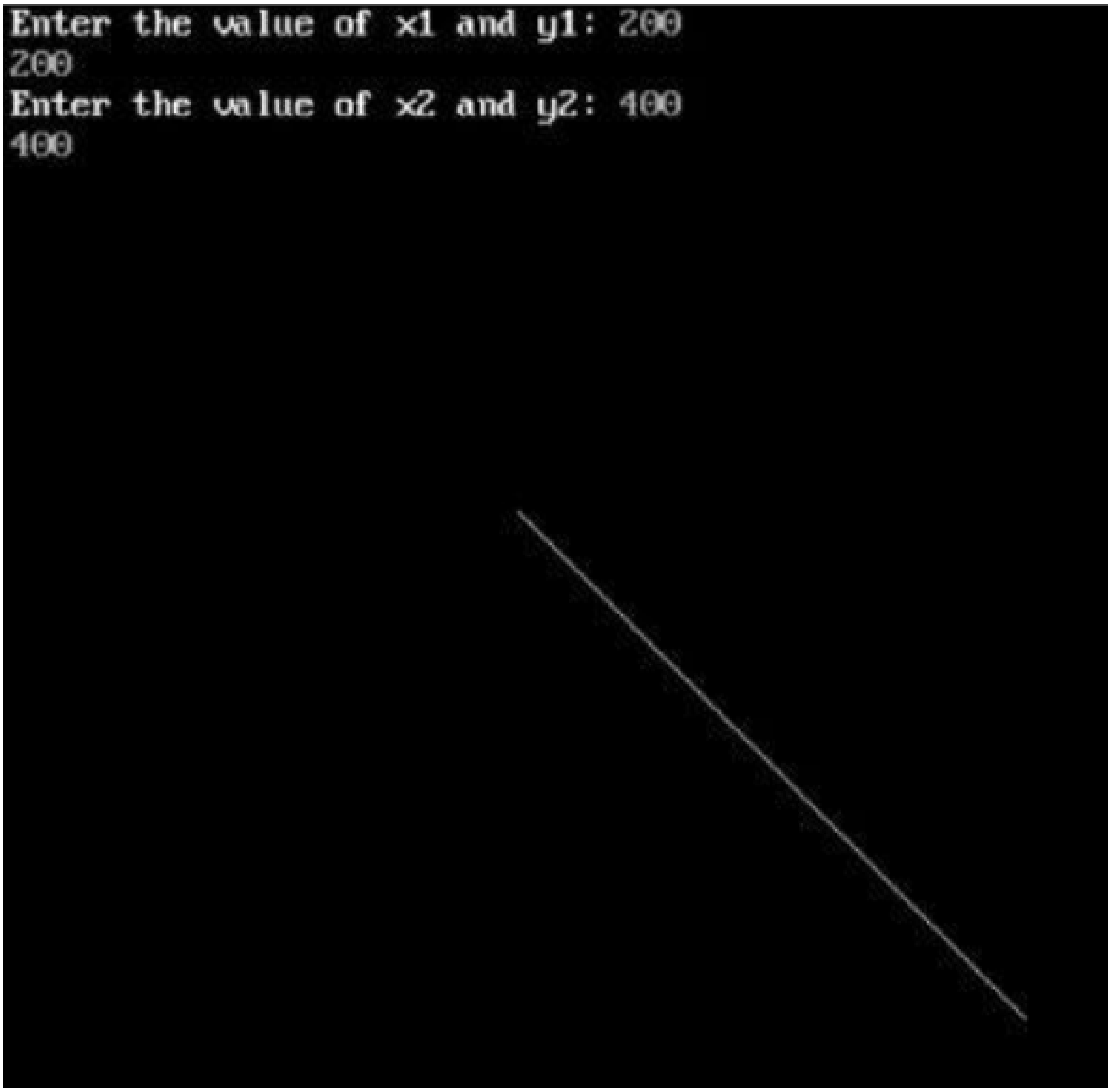
```
x=x+xinc;
putpixel(x,y,WHITE);
delay(20);
}
}
else
{
p=2*dx-dy;
while(y!=y2)
{
if(p<0)
{
p=p+2*dx;
}
```



```
}
else
{
p=p+2*dx-2*dy;
x=x+xinc;
if(p<0)
{
p=p+2*dx;
}
else{
p=p+2*dx-2*dy;
x=x+xinc;
x=x+xinc;
}
```



```
x=x+xinc;
x=x+xinc;
}
y=y+yinc;
y=y+yinc;
putpixel(x,y,WHITE);
delay(20);
}
}
getch();
closegraph();
restorecrtmode();
}
}
```



```
Enter the value of x1 and y1: 200
200
Enter the value of x2 and y2: 400
400
```

Conclusion: Thus, Bresenham's line drawing ~~of~~ algorithm is implemented successfully.