

Experiment No. 2Aim:

Implement generalized Bresenham's line drawing algorithm in C.

Algorithm:

Step 1: Read (x_1, y_1) and (x_2, y_2)

Step 2: Calculate $dx = |x_2 - x_1|$ and $dy = |y_2 - y_1|$ where $|a|$ is absolute value of a .

Step 3: Initialize $x = x_1$ and $y = y_1$.

Step 4: Calculate $s_1 = \text{sign}(x_2 - x_1)$ where sign is a function that returns $-1, 0$ & 1 and $s_2 = \text{sign}(y_2 - y_1)$.

Step 5: If $dy > dx$
then swap dy and dx and set $\text{exchange} = 1$
else set $\text{exchange} = 0$.

Step 6: Calculate initial decision parameter $e = 2dy - dx$.

Step 7: Set $i = 1$

Step 8: Plot (x, y)

Step 9: while $(e \geq 0)$

if $(\text{exchange} = 1)$

then $x = x + s_1$

else $y = y + s_2$

$e = e - 2dx$

Step 10: if $(\text{exchange} = 1)$

then $y = y + s_2$

else $x = x + s_1$

$e = e + 2dy$

Step 11: $i = i + 1$

Step 12: if $(i \leq dx)$ goto step 8.

PROGRAM:

Code:

```
#include <stdio.h>

#include <conio.h>

#include <graphics.h>

void Line(int, int, int, int);

void DashedLine(int, int, int, int);

void DottedLine(int, int, int, int);

void ThickLine(int, int, int, int);

int sign(int);

void main()

{

    int gd = DETECT, gm;

    int x1, y1, x2, y2, ch;

    initgraph(&gd, &gm, "C:\\TURBOC3\\BGI");

    printf("Enter the coordinates of the first point of the line\n");

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

    printf("Enter the coordinates of the second point of the line\n");

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

    printf("Enter the type of line you want to be drawn\n");

    printf("1-Solid\n2-Dotted\n3-Dashed\n4-Thick\n");

    scanf("%d", &ch);

    switch(ch)

    {

        case 1 :

            Line(x1, y1, x2, y2);

            break;

        case 2 :
```

```
DottedLine(x1, y1, x2, y2);
break;
case 3:
DashedLine(x1, y1, x2, y2);
break;
case 4:
ThickLine(x1, y1, x2, y2);
break;
default:
printf("INVALID INPUT!\n");
}
getch();
}
void Line(int x1, int y1, int x2, int y2)
{
int dx, dy, s1, s2, exchange, temp, e, i, x, y;
dx = abs(x2-x1);
dy = abs(y2-y1);
s1 = sign(x2-x1);
s2 = sign(y2-y1);
if(dy > dx)
{
exchange = 1;
temp = dx;
dx = dy;
dy = temp;
}
else
exchange = 0;
```

```

e = 2*dy-dx;
i = 1;
x = x1;
y = y1;
while(i <= dx)
{
    putpixel(x, y, 15);
    while(e >= 0)
    {
        if(exchange == 1)
            x += s1;
        else
            y += s2;
        e = e-2*dx;
    }
    if(exchange == 1)
        y += s2;
    else
        x += s1;
    e = e+2*dy;
    i++;
}

void DottedLine(int x1, int y1, int x2, int y2)
{
    int dx, dy, s1, s2, exchange, temp, e, i, x, y;
    dx = abs(x2-x1);
    dy = abs(y2-y1);
    s1 = sign(x2-x1);

```

```
s2 = sign(y2-y1);
if(dy > dx)
{
exchange = 1;
temp = dx;
dx = dy;
dy = temp;
}
else
exchange = 0;
e = 2*dy-dx;
i = 1;
x = x1;
y = y1;
while(i <= dx)
{
if(i % 2 == 0)
putpixel(x, y, 15);
while(e >= 0)
{
if(exchange == 1)
x += s1;
else
y += s2;
e = e-2*dx;
}
if(exchange == 1)
y += s2;
else
```

```

x += s1;
e = e+2*dy;
i++;
}
}

void DashedLine(int x1, int y1, int x2, int y2)
{
    int dx, dy, s1, s2, exchange, temp, e, i, x, y;
    dx = abs(x2-x1);
    dy = abs(y2-y1);
    s1 = sign(x2-x1);
    s2 = sign(y2-y1);
    if(dy > dx)
    {
        exchange = 1;
        temp = dx;
        dx = dy;
        dy = temp;
    }
    else
        exchange = 0;
    e = 2*dy-dx;
    i = 1;
    x = x1;
    y = y1;
    while(i <= dx)
    {
        if(i % 6 != 4 && i % 6 != 5)
            putpixel(x, y, 15);

```

```

while(e >= 0)
{
    if(exchange == 1)
        x += s1;
    else
        y += s2;
    e = e-2*dx;
}
if(exchange == 1)
    y += s2;
else
    x += s1;
e = e+2*dy;
i++;
}
}

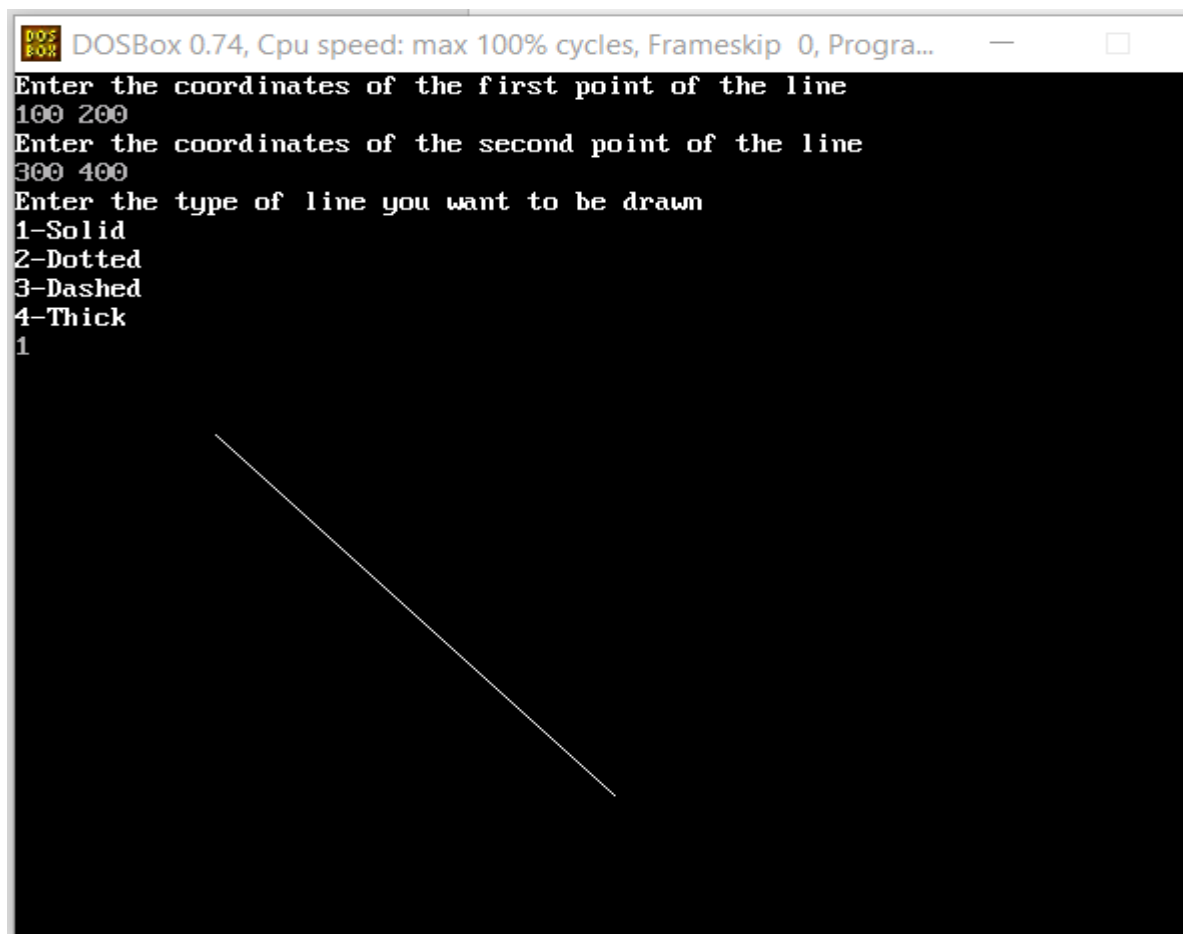
void ThickLine(int x1, int y1, int x2, int y2)
{
    Line(x1-1, y1, x2-1, y2);
    Line(x1, y1-1, x2, y2-1);
    Line(x1, y1, x2, y2);
    Line(x1+1, y1, x2+1, y2);
    Line(x1, y1+1, x2, y2+1);
}

int sign(int n)
{
    if(n > 0)
        return 1;
    else if(n < 0)

```

```
return -1;  
return 0;  
}
```

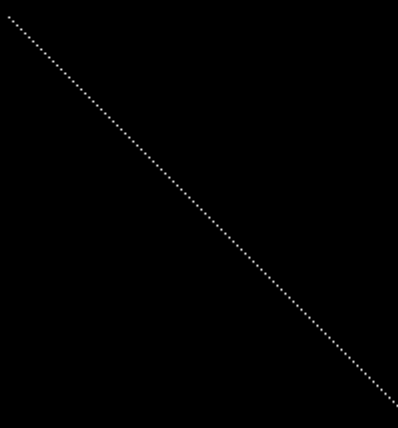
OUTPUT:



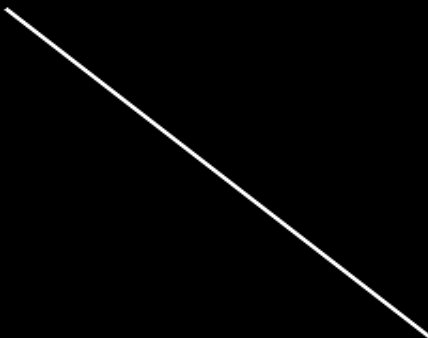
```
DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Progra...  
Enter the coordinates of the first point of the line  
100 200  
Enter the coordinates of the second point of the line  
300 400  
Enter the type of line you want to be drawn  
1-Solid  
2-Dotted  
3-Dashed  
4-Thick  
1  
100 200 300 400
```



```
DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Progra...
Enter the coordinates of the first point of the line
100 200
Enter the coordinates of the second point of the line
300 400
Enter the type of line you want to be drawn
1-Solid
2-Dotted
3-Dashed
4-Thick
2
```

A screenshot of a DOSBox window showing a line-drawing program. The user has entered the coordinates (100, 200) and (300, 400) and selected option 2 for a dotted line. The resulting image shows a black background with a dotted line extending from the top-left towards the bottom-right.

```
DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Progra...
Enter the coordinates of the first point of the line
100
200
Enter the coordinates of the second point of the line
300
400
Enter the type of line you want to be drawn
1-Solid
2-Dotted
3-Dashed
4-Thick
4
```

A screenshot of a DOSBox window showing the same line-drawing program. This time, the user has selected option 4 for a thick line. The resulting image shows a black background with a thick, solid white line extending from the top-left towards the bottom-right.



DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Progra...

Enter the coordinates of the first point of the line

400 300

Enter the coordinates of the second point of the line

200

100

Enter the type of line you want to be drawn

1-Solid

2-Dotted

3-Dashed

4-Thick

3

