

Name: Abhishek Singh Bisht

Course: B.C.A

Sec: A

Roll No: 1121005

Answer

A① DDA algorithm code

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

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

```
int main()
```

```
{
```

```
    int rou (float num)
```

```
    {
```

```
        return num < 0 ? num - 0.5 : num + 0.5;
```

```
    }
```

```
    int x1 = 100, x2 = 250, y1 = 100, y2 = 250, step;
```

```
    int gd = DETECT, gm;
```

```
    float x, y, m;
```

```
    int dx = x2 - x1;
```

```
    int dy = y2 - y1;
```

```
    m = dy / dx;
```

```
    if (dx > dy)
```

```
        step = dx;
```

```
    else
```

```
        step = dy;
```

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

```
    outtextxy (x1, y1, "A");
```

```
    putouttextxy (x2, y2, "B");
```

```
    putpixel (x1, y1, RED);
```

$x = x_1, y = y_1;$

while (step > 0)

{

if ($m < 1$)

{

$x = x + 1;$

$y = y + m;$

}

if ($m \geq 1$)

{

$x = x + 1/m;$

$y = y + 1;$

}

putpixel (rou(x), rou(y), RED);

}

getch();

return 0;

}

DDA Algorithm

- ① start
- ② Declare $x_1, y_1, x_2, y_2, dx, dy, x, y$ as integer variables.
- ③ Enter value of x_1, y_1, x_2, y_2 .
- ④ Calculate $dx = x_2 - x_1$
- ⑤ Calculate $dy = y_2 - y_1$
- ⑥ If $ABS(dx) > ABS(dy)$
Then step = abs(dx)
else

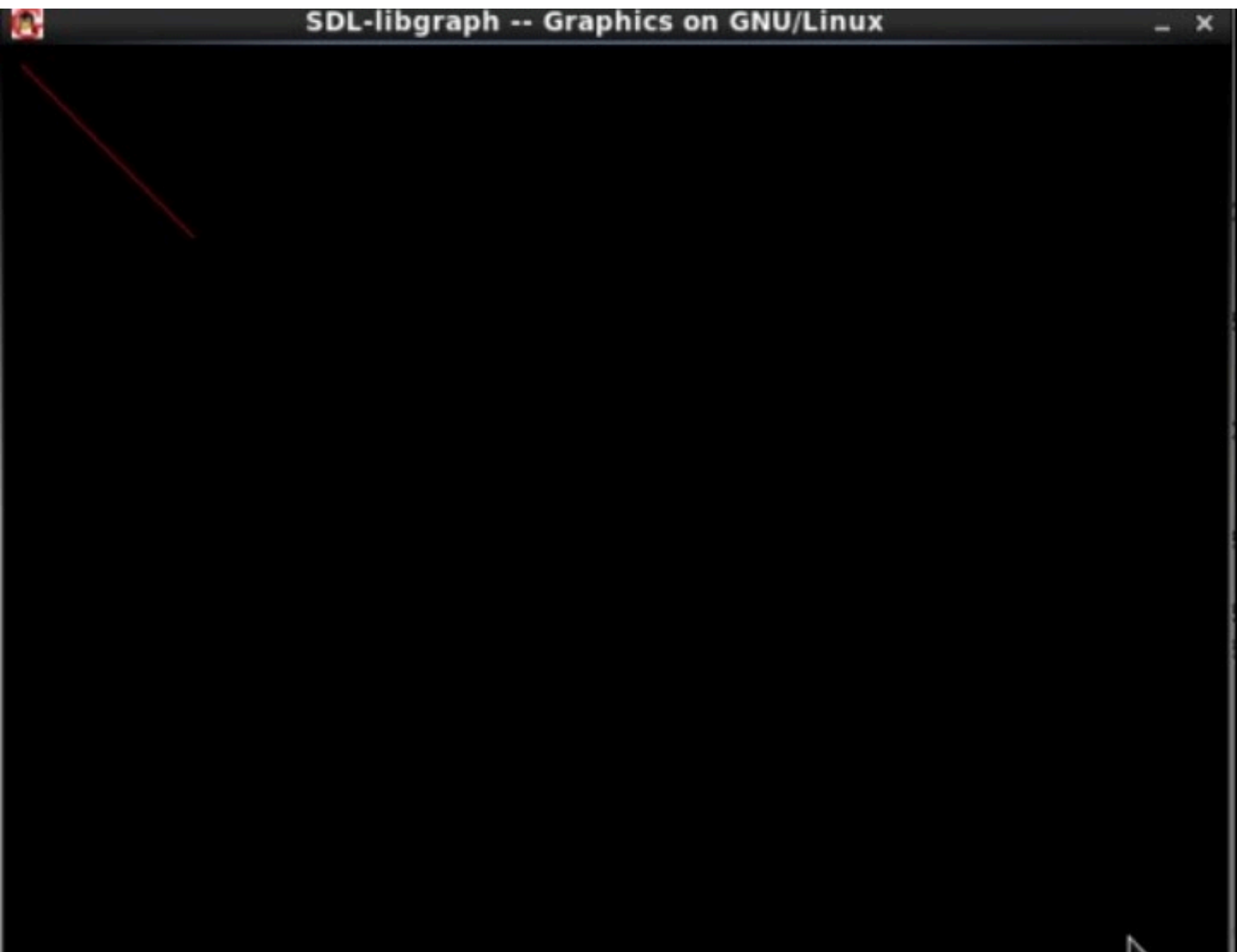
step ⑦ $x_{inc} = dx / step$
 $y_{inc} = dy / step$
assign $x = x_1$
assign $y = y_1$

step ⑧ set pixel (x, y)

step ⑨ $x = x + x_{inc}$
 $y = y + y_{inc}$
set pixels $(Round(x), Round(y))$

step ⑩ Repeat step ⑨ until $x = x_2$

step ⑪ End



Name: Abhishek Singh Bisht

Course: B.C.A

Sec: A

Roll NO: 1121005

Answer

A (3)

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

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

```
#include <stdlib.h>
```

```
main()
```

```
{
```

```
int gd = DETECT, gm, midx, midy;
```

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

```
midx = getmaxx() / 2;
```

```
midy = getmaxy() / 2;
```

```
setcolor (RED);
```

```
settextstyle (SCRIPT_FONT, HORIZ_DIR, 3);
```

```
settextjustify (CENTER_TEXT, CENTER_TEXT);
```

```
outtextxy (midx, midy - 10, "Traffic Light");
```

```
outtextxy (midx, midy + 10, "Press any key");
```

```
getch();
```

```
cleardevice();
```

```
setcolor (WHITE);
```

```
settextstyle (DEFAULT_FONT, HORIZ_DIR, 1);
```

```
rectangle (midx - 30, midy - 80, midx + 30, midy + 80);
```

```
circle (midx, midy - 50, 22);
```

```
setfillstyle (SOLID_FILL, RED);
```

```

floodfill (midx, midy - 50, WHITE);
setcolor (BLUE);
outtextxy (midx, midy - 50, "STOP");
delay (2000);
graphdefault 1();
cleardevice();
setcolor (WHITE);
rectangle (midx - 30, midy - 80, midx + 30, midy + 80);
circle (midx, midy, 20);
setfillstyle (SOLID_FILL, YELLOW);
floodfill (midx, midy, WHITE);
setcolor (BLUE);
outtextxy (midx - 18, midy - 3, "READY");
delay (2000);
cleardevice();
setcolor (WHITE);
rectangle (midx - 30, midy - 80, midx + 30, midy + 80);
circle (midx, midy + 50, 22);
setfillstyle (SOLID_FILL, GREEN);
floodfill (midx, midy + 50, WHITE);
setcolor (BLUE);
outtextxy (midx - 7, midy + 48, "Go");
setcolor (RED);
settextstyle (SCRIPT_FONT, HORIZ_DIR, 4);
outtextxy (midx - 150, midy + 100, "Press any key");
getch();
closegraph();
return 0;
}

```



