

Ans 1  $\Rightarrow$  ① Algorithm to implement 8-connected flood  
fill.

Step 1 - Start Algorithm

Step 2 - Draw the rectangle using rectangle function.

Step 3 - Implement 8 connected flood fill with the  
co-ordinates  $x$  and  $y$ .

put pixel ( $x, y, \text{new col}$ );

flood fill ( $x+1, y, \text{old}, \text{new col}$ );

flood fill ( $x-1, y, \text{old}, \text{new col}$ );

flood fill ( $x, y+1, \text{old}, \text{new col}$ );

flood fill ( $x, y-1, \text{old}, \text{new col}$ );

flood fill ( $x+1, y+1, \text{old}, \text{new col}$ );



flood fill (  $x-1$ ,  $y+1$ , old, new col )

flood fill (  $x+1$ ,  $y-1$ , old, new col )

flood fill (  $x-1$ ,  $y-1$ , old, new col )

Step 4 - Stop Algorithm.



© Program :

➔ # include < stdio.h >

# include < graphics.h >

# include < dos.h >

# include < conio.h >

void flood fill ( int x , int y , int old , int new col )  
{

int current ;

current = getpixel ( x , y ) ;

if ( current == old )

{

delay ( 5 ) ;

putpixel ( x , y , new col ) ;

flood fill ( x + 1 , y , old , new col ) ;

flood fill ( x - 1 , y , old , new col ) ;



```
floodfill (x, y+1, old, new_col);
```

```
floodfill (x, y-1, old, new_col);
```

```
floodfill (x+1, y+1, old, new_col);
```

```
floodfill (x-1, y+1, old, new_col);
```

```
floodfill (x+1, y-1, old, new_col);
```

```
floodfill (x-1, y-1, old, new_col);
```

```
}
```

```
}
```

```
void main ()
```

```
{
```

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

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

```
rectangle (50, 50, 150, 150);
```

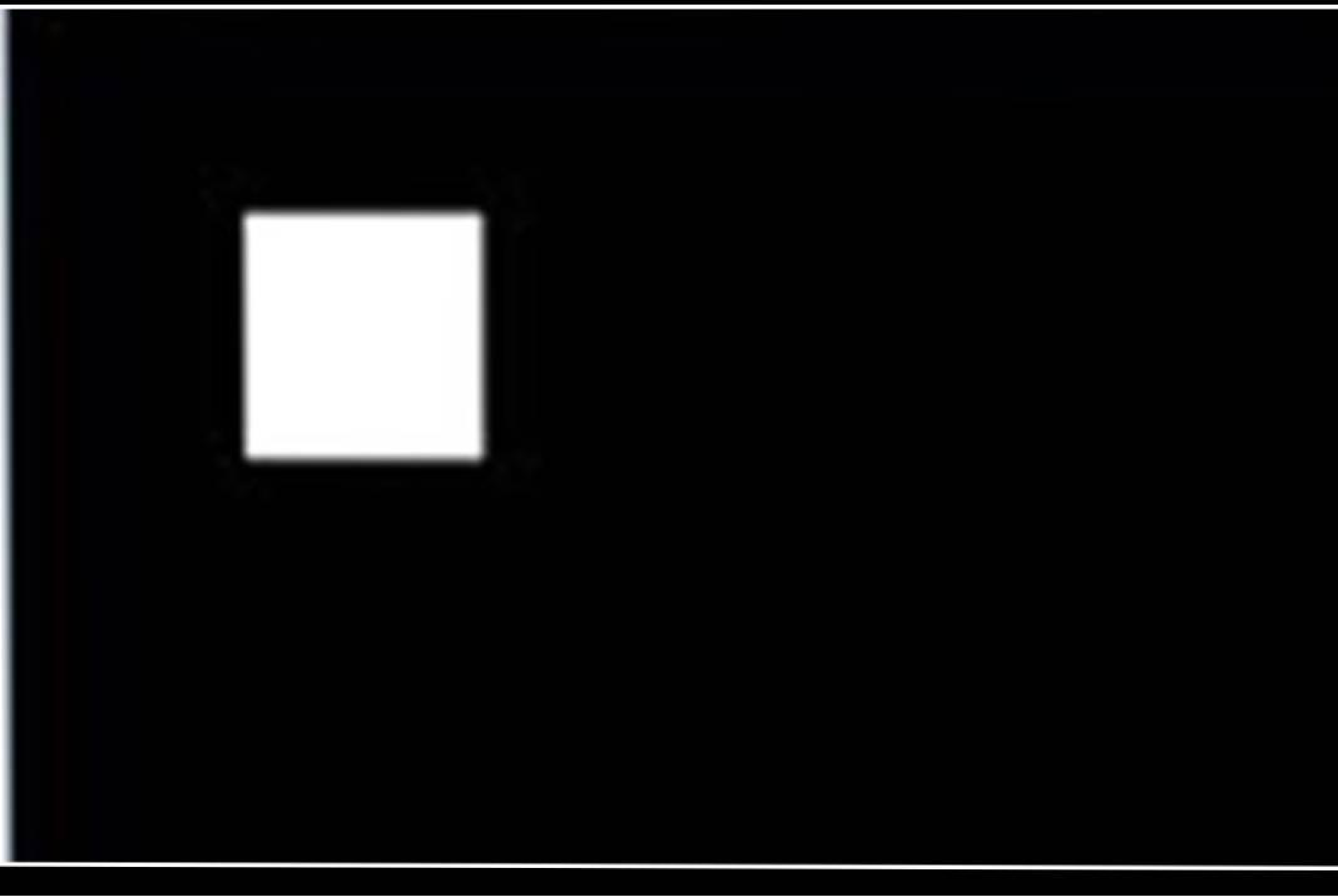
```
floodfill (70, 70, 0, 15);
```

```
getch ();
```

```
closegraph ();
```

```
}
```







# ① Computer Graphics Practical

Om Sah

1121096 (15)

BCA 6<sup>th</sup> B

Ans 3 ⇒ ① Bresenham's Circle Algorithm :-

Step 1 - Start Algorithm.

Step 2 - Declare  $p, q, x, y, r, d$  variables

$p, q$  are co-ordinates of the center of the circle  
 $r$  is radius of circle.

Step 3 - Enter the value of  $r$ .

Step 4 - Calculated  $d = 3 - 2r$

Step 5 - Initialize  $x = 0$

&  $y = r$

Step 6 - Check if the whole circle is scan converted  
If  $x \geq y$   
Stop



Step 7 - Plot eight points by using concepts of eight-way symmetry. The center is at  $(p, q)$ . Current active pixel is  $(x, y)$ .

putpixel  $(x+p, y+q)$

putpixel  $(y+p, x+q)$

putpixel  $(-y+p, x+q)$

putpixel  $(-x+p, y+q)$

putpixel  $(-x+p, -y+q)$

putpixel  $(-y+p, -x+q)$

putpixel  $(y+p, -x+q)$

putpixel  $(x+p, -y+q)$

Step 8 - Find location of next pixels to be

scanned.

If  $d < 0$

then  $d = d + 4x + 6$



Increment  $x = x + 1$

If  $d \geq 0$

then  $d = d + 4(x - y) + 10$

Increment  $x = x + 1$

decrement  $y = y - 1$

Step 9 - Go to Step 6

Step 10 - Stop Algorithm.

① Program :

> #include <graphics.h>

#include <stdlib.h>

#include <stdio.h>

#include <conio.h>

#include <math.h>



```
Void Eightway Symmetric Plot (int xc, int yc, int x,  
int y)
```

```
{
```

```
putpixel (x + xc, y + yc, RED);
```

```
putpixel (x + xc, -y + yc, YELLOW);
```

```
putpixel (-x + xc, -y + yc, GREEN);
```

```
putpixel (-x + xc, y + yc, YELLOW);
```

```
putpixel (y + xc, x + yc, 12);
```

```
putpixel (y + xc, -x + yc, 14);
```

```
putpixel (-y + xc, -x + yc, 15);
```

```
putpixel (-y + xc, x + yc, 6);
```

```
}
```

```
Void Bresenham Circle (int xc, int yc, int r)
```

```
{
```

```
int x = 0, y = r, d = 3 - (2 * r);
```



Eight way Symmetric Plot ( $x_c, y_c, x, y$ );

while ( $x \leq y$ )

{

if ( $d \leq 0$ )

{

$d = d + (4 * x) + 6;$

}

else

{

$d = d + (4 * x) - (4 * y) + 10;$

$y = y - 1;$

}

$x = x + 1;$

Eight way Symmetric Plot ( $x_c, y_c, x, y$ );

}

}

int main (void)

int  $x_c, y_c, x, y$ ,  $gdriver = DETECT$ ,  $gmode$ ,  $errcode$ ;



```
init_graph (& gdriver, & gmode, "n");
```

```
errorcode = graph_result ();
```

```
if (errorcode != 0)
```

```
{
```

```
printf ("Graphics error: %d\n", graph_err_msg  
        (errorcode));
```

```
printf ("Press any key to halt");
```

```
getch ();
```

```
exit (1);
```

```
}
```

```
printf ("Enter values of xc and yc: ");
```

```
scanf ("%d %d", & xc, & yc);
```

```
printf ("Enter values of radius: ");
```

```
scanf ("%d", & r);
```

```
Bresenham_circle (xc, yc, r);
```

```
getch ();
```

```
close_graph ();  
return 0; }
```



Enter the values of xc and yc :100 100  
Enter the value of radius :50

