

Nikhil Thapa

1121092 (12).

BCA 6(B)

Ans-1

Algorithm :-

Step 1: Start

Step 2: Draw the rectangle using rectangle function.

Step 3: Implement 8 connected floodfill with the coordinates x and y .

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

Step 4: Stop.

program

#include <stdio.h>

#include <graphics.h>

#include <conio.h>

void floodfill(int x, int y, int old, int newcol)

{

int current;

current = getpixel(x, y);

if (current == old)

{

delay(5);

putpixel(x, y, newcol);

floodfill(x+1, y, old, newcol);

floodfill(x-1, y, old, newcol);

floodfill(x, y+1, old, newcol);

floodfill(x, y-1, old, newcol);

floodfill(x+1, y+1, old, newcol);

floodfill(x-1, y+1, old, newcol);

floodfill(x+1, y-1, old, newcol);

floodfill(x-1, y-1, old, newcol);

}


```
void main()
```

```
{
```

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

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

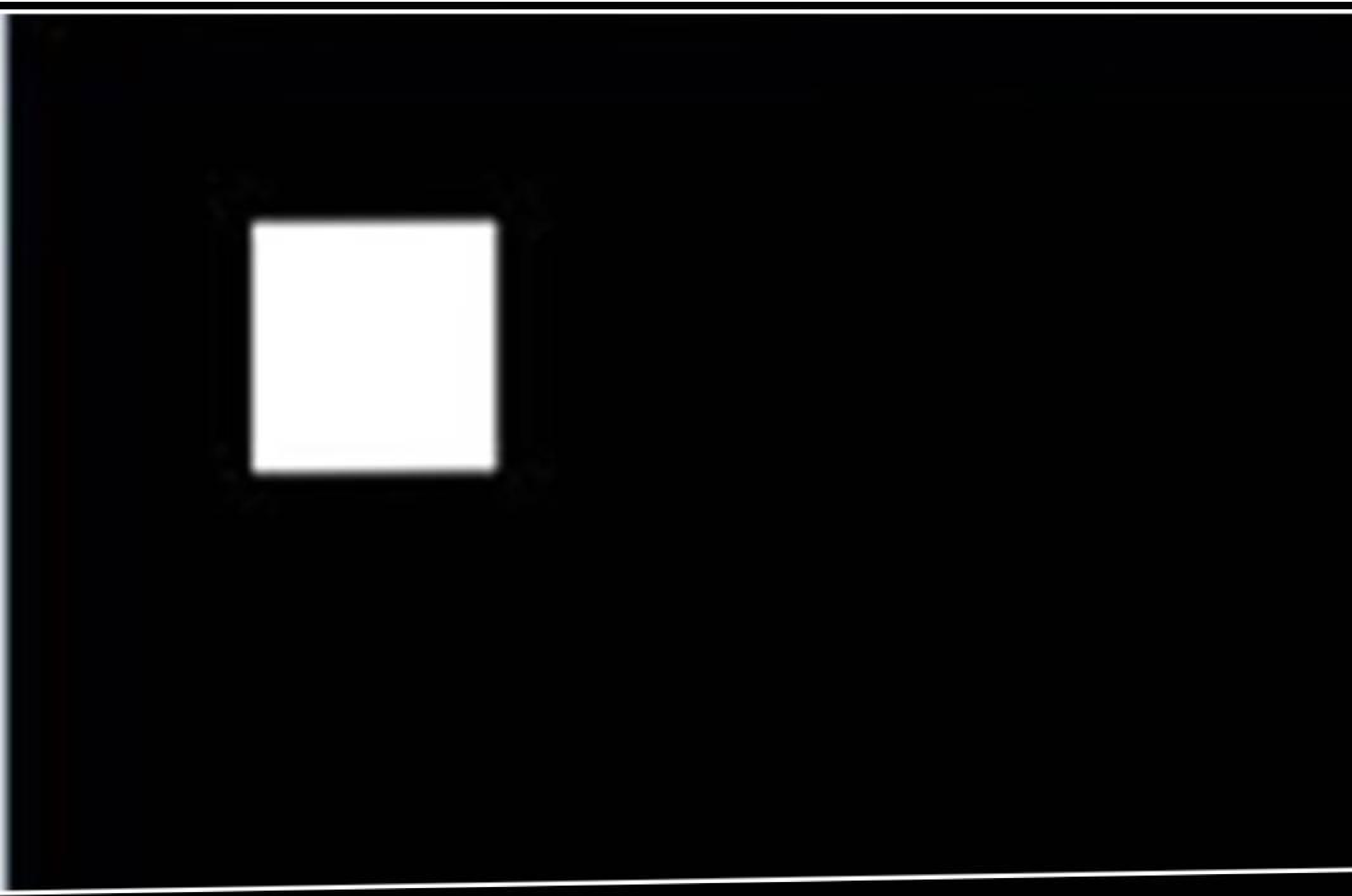
```
    rectangle(50, 50, 150, 150);
```

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

```
    getch();
```

```
    closegraph();
```

```
}
```



Nikhi Thapa

1121092(12)

BCA 6(B)

Computer Graphics & Animation.

Ans-3 Algorithm:-

Step 1:- Start-

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

p, q are coordinates of the center of the circle

r is the radius of circle.

Step 3:- Enter the value of r .

Step 4:- Calculate $d = 3 - 2r$

Step 5:- Initialize $x = 0$ & $assy = r$

Step 6:- Check if the whole circle is scan converted.

if $x \geq y$

Stop.

Step 7:- Plot eight points by using concept of eight-way

symmetry. The center is at (p, q) . Current active

pixel is (x, y) .

$\text{putpixel}(x+p, y+q)$
 $\text{putpixel}(y+p, x+q)$
 $\text{putpixel}(-y+p, x+q)$
 $\text{putpixel}(-x+p, y+q)$
 $\text{putpixel}(-x+p, -y+q)$
 $\text{putpixel}(-y+p, -x+q)$
 $\text{putpixel}(y+p, -x+q)$
 $\text{putpixel}(x+p, -y-q)$.

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 Alg.

program to draw a circle using Bresenham's circle drawing algorithm.

#include <graphics.h>

#include <stdlib.h>

#include <stdio.h>

#include <conio.h>

#include <math.h>

void EightWaySymmetricPlot (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 BresenhamCircle (int xc, int yc, int r)
```

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

```
EightWaySymmetricPlot (xc, yc, x, y);
```

```
while (x <= y)
```

```
{ if (d <= 0)
```

```
{
```

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

```
}
```

```
else
```

```
{
```

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

```
y = y - 1;
```

```
}
```

```
x = x + 1;
```

```
EightWaySymmetricPlot (xc, yc, x, y);
```

```
}
```

```
}
```

```
int main (void)
```

```
{
```

```
int xc, yc, r, gdriver = DETECT, gmode, errorcode;
```



```

initgraph (&gdvuer, &gmode, " ");
errorcode = graphresult();
if (errorcode != 0) {
    printf ("Graphic error: %s\n", graphErrorMsg(errorcode));
    printf ("Press any key to halt:");
    getch();
    exit(1);
}
printf ("Enter the value of xc and yc:");
scanf ("%d%d", &xc, &yc);
printf ("Enter the value of radius:");
scanf ("%d", &r);
BresenhamCircle(xc, yc, r);
getch();
closegraph();
return 0;
}

```


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

