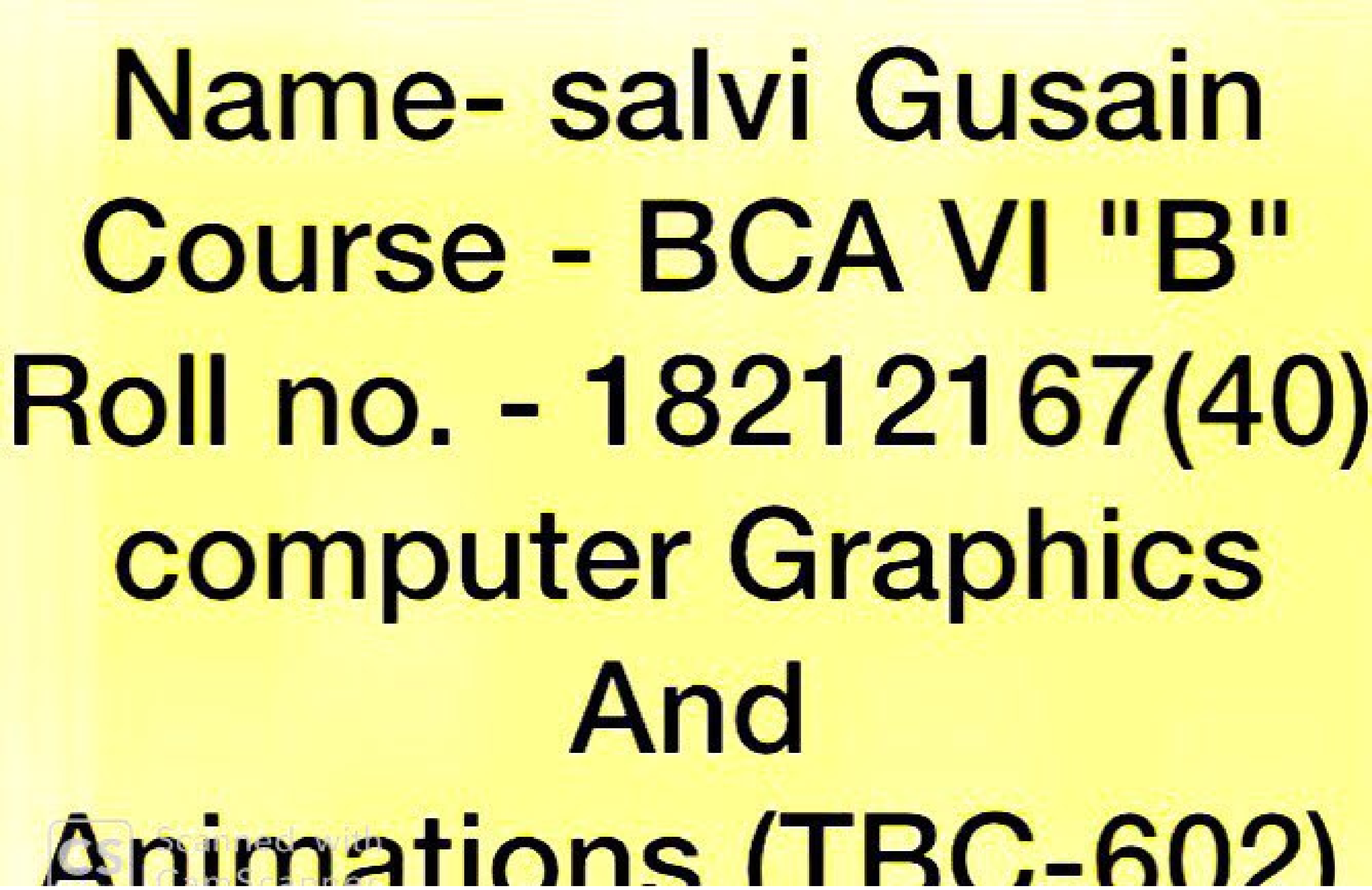


Name- salvi Gusain
Course - BCA VI "B"
Roll no. - 18212167(40)
computer Graphics
And
Animations (TBC-602)

Salvi Gusain
BCA 6th "B"
40

Ans1.

8 - connected Flood -Fill Algorithm -

Algorithm -

⇒ In thus, a point or seed which is inside region is selected. Ths point is called a seed point then 4 connected approaches or eight connected approaches is used to fill with spegfied color

```
{
 void floodfill (x,y, fill_color, old-color, integer)
    if (getpixel(x,y) = old- color)
  {
  Setpixel (x,y, fill-color);
  fill (x+1,y, fill_color, old_color);
  fill (x-1, y, fill-color, old -color);
  fill (x, y+1, fill-color, old-color);
  fill (x, y-1, fill_color, old -color);
  }
  }
```

Name- Salvi Gusain

Course- BCA 6th "B"

Roll no - 40 (18212167)

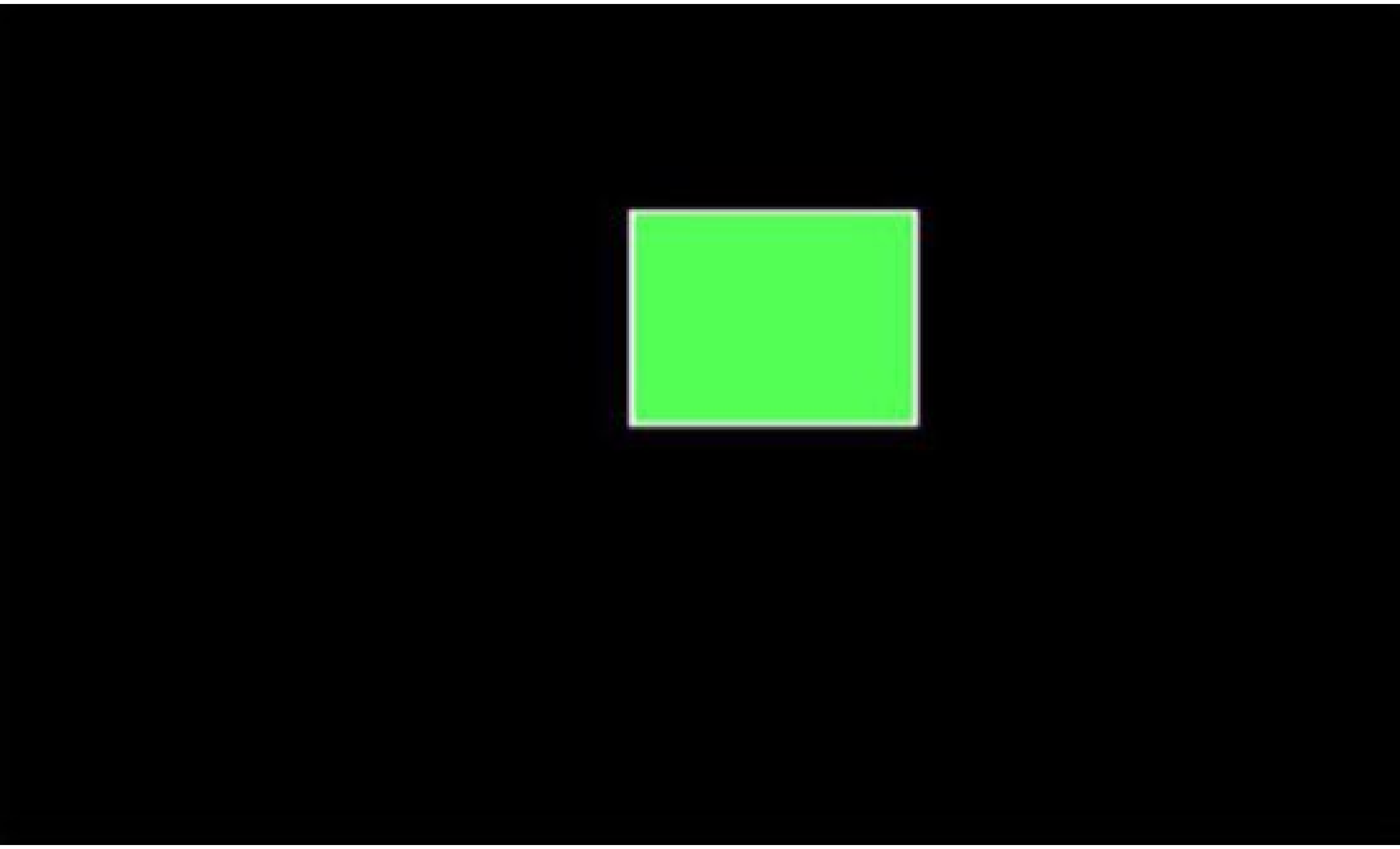
Pratical of Computer Graphic (PBC-602)

hs1 - 8- Connected Flood-Fill Algorithm-

Program

```
#include <stdio.h>
# include <graphics.h>
# include <dos.h>
# include <conio.h>
void floodfill (int x, int y, intold, intnewcol)
{
    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();
}
```

output



Ans3 - Algorithm and Program to implement Bresenham Circle Drawing Algorithm -

Algorithm -

Step1 - Start Algorithm

Step2 - Declare p, q, ~~x~~ x, y, r, d variables p and q are coordinate of the center of circle.

Step3 - Enter the value of r

Step4 - Calculate d = 3-2r

Step5 - Initialize x=0 and nbsy = r

Step6 - Check the whole circle is Scan converted if x>=y , stop.

Step7 - Plot eight point by using concepts of eight way symmetry.

The center is at (p,q). current pixel (x, y)

Putpixel (Y+p , y+q)

putpixel (y+p, x+p)

put pixel (-y+p , x+p)

put pixel (-x+p , Y+p)

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

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

put pixel (y+p, -x+q)

putpixel (x+p, -y-q)

Kdv Gusnah

Step8- 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 $y = y+1$

Step9 - Repeat the Step 6

Step10 - Stop Algorithm

Program -

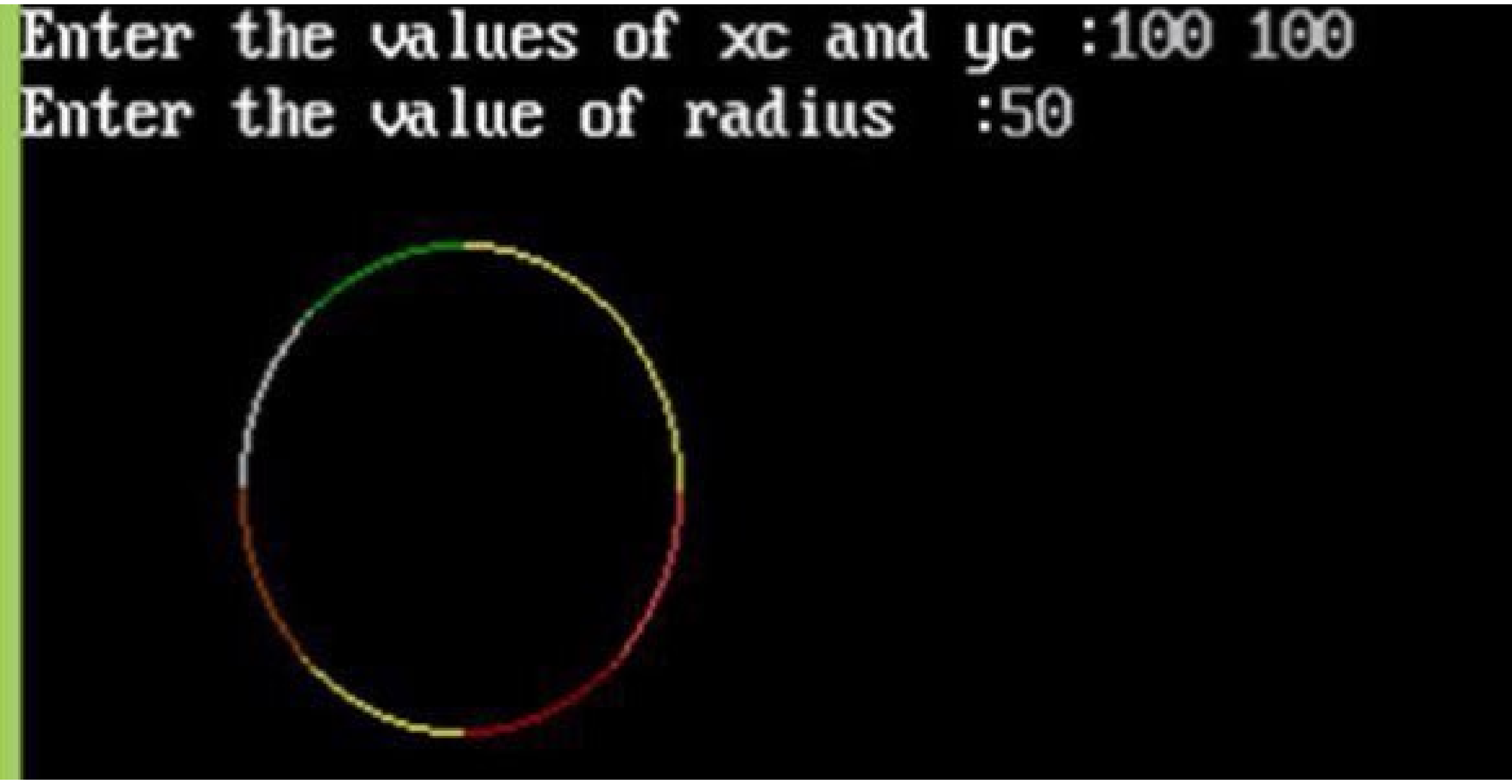
```
#include <graphics.h>
# include < Stdlib.h>
# include < Stdio.h>
# include < conio.h>
# include < math.h>

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+y c, 12);
putpixel (y + xc, -x+y c, 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 SymmetricPlot (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;
Eight way Symmetric Plot (xc,yc,x,y);
}
}
 int main (void)
 {
int xc, yc, r, gdriver = DETECT, gmode, errorcode;
 initgraph (&gdriver, &gmode,);
errorcode = graphresult();
 if (errorcode != grok)
```

```
{
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);
Bresenham Circle (xc,yc,r);
getch();
closegraph();
return0;
}
```

Output-

```
Enter value of xc and yc - 100  100
Enter value of radius   - 50
```



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