END TERM PRACTICAL EXAMINATIONS.

Name :- Rubiya Rawat

33 Class Roll no.

father's Name :- Late M.P.S Rawat

University Roll no. :- 1121114

Course :- BCA 'B'

Semester :- 6

Paper Name :- Computer Graphics and Animation Practical.

Paper Code :- PBC 602.

Type of Paper :- Regular.

Date of Examination :- 16th June 2021.

SET-B

Ques) Flood fill algorithm with 8 connected Approach

```
#include<stdio.h>
#include<graphics.h>
#include<dos.h>
void floodfill(int a, int b, int x, int y)
 {
   int current;
   current=getpixel(a,b);
   if(current==x)
    {
     delay(5);
     putpixel(a,b,y);
     floodfill(a+1,b,x,y);
     floodfill(a-1,b,x,y);
     floodfill(a,b+1,x,y);
     floodfill(a+1,b+1,x,y);
     floodfill(a-1,b+1,x,y);
     floodfill(a+1,b-1,x,y);
     floodfill(a-1,b-1,x,y);
    }
 }
void main()
{
   int gd=DETECT,gm;
   initgraph(&gd,&gm,"");
   rectangle(50,50,150,150);
```

Rukiya

```
    floodfill(70, 70, 0, 15);
    getch( );
    closegraph();
}.
```

Kubiya.

Algorithm

Step 01 :- Start.

Step 02 :- Initialize the value of (centre) seed point. (a, b), fill color = x and old color = y.

Step 03 :- Define the boundary values of the polygon.

Step 04 :- Check if the current centre point is of default color then repeat step 05 and step 06 till the boundary pixel is reached.

Step 05 :- Change the default color with the fill color (x) at the centre (a, b).

Step 06 :- Recursively follow the procedure with four neighbourhood points—.

```
floodfill (a+1, b, x, y)
floodfill(a-1, b, x, y)
floodfill (a, b+1, x, y).
floodfill (a+1, b+1, x, y)
floodfill (a-1, b+1, x, y)
floodfill (a+1, b-1, x, y)
floodfill (a-1, b-1, x, y).
```

Step 07 :- Stop.

Rukiya.

SET-B.

Q3 Bresenham's Circle Drawing algorithm.

```
#include <stdio.h>
#include<graphics.h>
int main()
{
int gd=DETECT, gm;
int r, x, y, p, xc=200, yc=200;
printf ("Enter radius");
scanf ("%d", &r);
initgraph(&gd, &gm, "");
x=0;
y=r;
p=3-(2*r);
for (x=0; x<=y; x++)
{
    if (p<0)
    {
        y=y;
        p=p+(4*x)+6;
    }
    else
    {
        y=y-1;
        p=p+(4*x)-(4*y)+10;
        x=x+1;.
    }
putpixel(xc+x, yc+y, 1);
putpixel(xc+y, yc+x, 2);
putpixel (xc-x, yc+y, 3);
putpixel ( xc-y, yc+x, 4);
putpixel (xc-x, yc-y, 5);
putpixel (xc-y, yc-x, 6);
putpixel (xc+x, yc-y, 7);
putpixel (xc+y, yc-x, 8);
}
getch();
closegraph();
return 0;
}.
```

Rubiya.

Algorithm

```
step01 :- Start
step02 :- Declare r, x, y, p, xc, yc variables.
    → xc and yc are the coordinates of the center of the
      circle.
    → r is the radius.
step 03 :- Enter the value of r.
step04 :- calculate p=3-(2*r);
step05 :- Initialize x=0 and y=r;
step06 :- Check if the whole circle is scan converted.
    If x>=y
        stop.
step07 :- Plot eight points by using concepts of eight-
    way symmetry. The centre at (xc, yc). Current
    active pixel is (x,y)..

    putpixel (xc+x, yc+y, 1).
    putpixel (xc+y, yc+x, 2)
    putpixel (xc-x, yc+y, 3)
    putpixel (xc-y, yc+x, 4)
    putpixel (xc-x, yc-y, 5)
    putpixel (xc-y, yc-x, 6)
    putpixel (xc+x, yc-y, 7)
    putpixel (xc+y, yc-x, 8).
```

Rubiya.

```
Step08 :- find location of next pixels to be scanned
    if $P<0$.
        then $p=p+4x+6$.
    no increment in y.

    if $p>=0$.
        then $p=p+4x-(4-y)+10$.
        decrement $y=y-1$.
        increment $x=x+1$.

step09 :- Go to step 6.

step10 :- Stop.
```