# COMPUTER GRAPHICS

## END SEMESTER PRACTICAL

Name — Saloni Maggo

Course — BCA 6 B

Roll no — 39

University roll no. — 1121120

Exam — Computer Graphics End Sem Practical Examination

Email — salonimaggo2001@gmail.com

P3 — Bresenham circle drawing algorithm.

Source code —

```c
#include <stdio.h>
#include <graphics.h>
void main()
{
    int gd = DETECT, gm;
    int n, x, y, p, xc = 320, yc = 240;
    printf("Enter the radius");
    scanf("%d", &n);
    initgraph(&gd, &gm, " ");
    x = 0;
    y = n;
    putpixel(xc+x, yc-y, 1)
    p = 3 - (2*n)
    for(x = 0; x <= y; x++)
    {
        if(p < 0)
        {
            y = y;
            p = (p + (4*x) + 6);
        3
```

3

```
else
{
  y= y-1;
  p= p+ ((4* (x-y)+10));
}
  putpixel ( xc+x, yc-y , 1);

  putpixel ( xc-x , yc-y , 2);

  putpixel ( xc+x , yc+y , 3);

  putpixel ( xc-x , yc+y , 4);

  putpixel ( xc+y , yc-x , 5);

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

  putpixel( xc+y , yc+x , 7);

  putpixel ( xc-y , yc+x , 8);
}

getch();

closegraph();
}
```

# Algorithm –

Step 1 – Start.

Step 2 – Declare integer variables. $r, x, y, p, xc = 320$ and $yc = 240$. where $r$ is the radius of the circle.

Step 3 – Enter the value of radius.

Step 4 – Initialize $x = 0$ and $y = r$.

Step 5 – Plot putpixel $(xc + x, yc - y; 1)$

Step 6 – Calculate $p = 3 - 2r$.

Step 7 – Check if the whole circle is scan converted if $x >= y$ stop.

Step 8 – Plot eight points by using concept of eight way symmetry, current active pixel is $x, y$. center is at $(a, b)$

putpixel $(x + a, y + b)$

putpixel $(y + a, x + b)$

putpixel $(-y + a, x + b)$

putpixel $(-x + a, y + b)$

putpixel $(-x + a, -y + b)$

putpixel $(-y + a, -x + b)$

putpixel $(y + a, -x + b)$

putpixel $(x + a, -y - b)$

Step 9 — find location of next pixels to be scanned

$$\text{if } p < 0$$

$$\text{then } y = y$$

$$p = p + 4x + 6$$

use

$$p = p + 4(x - y) + 10$$

$$\text{decrement } y = y - 1$$

Step 10 — Go to step 7.

Step 11 — Stop.

P2—

2D transformation of Triangle

```c
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>
#include <math.h>
void main()
{
    int gm;
    int gd = DETECT;
    int x1, x2, x3, y1, y2, y3, nx1, nx2, nx3, ny1, ny2, ny3, c;
    int sx, sy, xt, yt, n;
    float t;
    initgraph(&gd, &gm, "");
    printf("\t Program for basic transaction ");
    printf("\n \t Enter points of triangle ");
    setcolor(1);

    scanf("%d %d %d %d %d %d", &x1, &y1, &x2, &y2,
                                    &x3, &y3);
    line(x1, y1, x2, y2);
```

```
line ( x2, y2, x3, y3 );

line ( x3, y3, x1, y1 );

getch ();

printf (" \n 1. Transaction \n 2. Rotation \n 3. Scalling \n
                4. Exit ");

printf (" Enter ur choice ");

scanf (" %d ", &c);

switch ( c )
{
    case 1:   printf (" \n Enter translation factor ");

              scanf (" %d %d ", &xt, &yt);

              nx1 = x1 + xt;
              ny1 = y1 + yt;
              nx2 = x2 + xt;
              ny2 = yt + y2;
              nx3 = x3 + xt
              ny3 = y3 + yt;

              line ( nx1, ny1, nx2, ny2 );
              line ( nx2, ny2, nx3, ny3 );
              line ( nx3, ny3, nx1, ny1 );
              getch ();
```

Case 2:

```
printf ("\n Enter the angle of rotation ");
scanf ("%d", &n);
t= 3.14 * n / 180;
nx1 = abs ( x1 * cos (t) - y1 * sin(t));
ny1 = abs ( x1 * sin(t) + y1 * cos (t));
nx2 = abs ( x2 * cos(t) - y2 * sin(t));
ny2 = abs ( x2 * sin (t) + y2 * cos(t));
nx3 = abs ( x3 * cos(t) - y3 * sin(t));
ny3 = abs ( x3 * sin(t) + y3 * cos(t));
line (nx1, ny1, nx2, ny2);
line ( nx2, ny2, nx3, ny3);
line ( nx3, ny3, nx1, ny1);
getch();
```

Case 3:

```
printf ("\n Enter scaling factor ");
scanf (" %d %d ", &sx, &sy);
nx1 = x1 * sx;
ny1 = y2 * sy;
nx2 = x2 * sx;
ny2 = y2 * sy;
nx3 = x3 * sx;
```

```
ny3 = y3 * sy;
line ( nx1, ny1, nx2, ny2);
line ( nx2, ny2, nx3, ny3);
line ( nx3, ny3, nx1, ny1);
getch();

Case 4:
        break;

default:
        printf (" Enter the correct choice ");

    3

    closegraph ();

    3
```