**Experiment No.-5**

**Student Name: PUJA KUMARI UID: 20BCA1448**

**Branch: BCA Section/Group: 20BCA5-B**

**Semester: 5th Date of Performance: 21.10.22**

**Subject Name: COMPUTER GRAHICS LAB Subject Code: 20CAP-316**

1. **Aim/Overview of the practical:** WAP to rotate a line using points a (100,100) and B (100,200) by rotation angle 45 degree.
2. **Task to be done:** To rotate a line using points a (100,100) and B (100,200) by rotation angle 45 degree.
3. **Concept used:**

* The point (x, y) is to be rotated
* The (xc yc) is a point about which counter clockwise rotation is done
* **Step1:** Translate point (xc yc) to origin
* **Step2:** Rotation of (x, y) about the origin.
* **Step3:** Translation of center of rotation back to its original position.

1. **Steps/Commands involved to perform practical:**

#include <conio.h>

#include <graphics.h>

#include <math.h>

#include <stdio.h>

#include <stdlib.h>

using namespace std;

// Ellipse drawing function

void drawEllipse(int xc, int yc, int a, int b,

float alpha, int color)

{

float t = 3.14 / 180;

alpha = 360 - alpha;

setcolor(color);

int theta;

// Filling each pixel corresponding

// to every angle from 0 to 360

for (int i = 0; i < 360; i += 1) {

theta = i;

int x = a \* cos(t \* theta) \* cos(t \* alpha)

+ b \* sin(t \* theta) \* sin(t \* alpha);

int y = b \* sin(t \* theta) \* cos(t \* alpha)

- a \* cos(t \* theta) \* sin(t \* alpha);

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

}

}

// Function to calculate the position

// of ellipse after each rotation

void slidePattern(int xc, int yc, int r, int a, int b,

int alpha, float p, int color)

{

setcolor(color);

float t = 3.14 / 180;

float t1, t2, d;

float angle = (p \* alpha);

// Calculation for center of Ellipse

t1 = cos(t \* fmod(angle, 360));

t2 = sin(t \* fmod(angle, 360));

t1 \*= t1;

t2 \*= t2;

t1 = t1 / (a \* a);

t2 = t2 / (b \* b);

d = sqrt(t1 + t2);

d = 1 / d;

int draw\_x = xc + (r + d) \* cos(t \* alpha);

int draw\_y = yc - (r + d) \* sin(t \* alpha);

int draw\_ang = angle + alpha;

drawEllipse(draw\_x, draw\_y, a,

b, draw\_ang, color);

}

// Function to increment the angle

// of rotation

void ellipseovercircle(int xc, int yc,

int r, int a, int b)

{

float theta = 0;

double h, p1;

// Calculating the ratio of

// perimeters of Ellipse and Circle

h = (a \* a) + (b \* b);

h /= 2;

p1 = sqrt(h);

p1 /= r;

p1 = 1 / (p1);

// by decreasing theta we can

// move Ellipse clockwise

for (;; theta -= 1) {

// Draw Ellipse at new location

// using White color

slidePattern(xc, yc, r, a, b,

theta, p1, WHITE);

circle(xc, yc, r); // Drawing Circle

delay(25); // Introducing delay

// Erase the existing Ellipse

slidePattern(xc, yc, r, a, b,

theta, p1, BLACK);

}

}

// Driver code

int main()

{

// Initialize graphics function

int gd = DETECT, gm;

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

// maximum X-coordinate for the window

int maxx = getmaxx();

// maximum Y-coordinate for the window

int maxy = getmaxy();

// Start drawing from the mid of the screen

ellipseovercircle(maxx / 2, maxy / 2,

100, 40, 28);

closegraph();

return 0;

}

**5. Result/Output/Writing Summary:**

