**Answer Qusetion no:1**

// Sample Line Incremental (DDA) Algorithm

#include<windows.h>

#include<iostream>

#include<math.h>

#include <stdio.h>

#include<GL/gl.h>

#include <GL/glut.h>

using namespace std;

int X1, Y1, X2, Y2, r ,cx, cy;

void DDA(void)

{

double dx=(X2-X1),dy=(Y2-Y1),l;

float xInc,yInc,x=X1,y=Y1;

int ds=2\*dy - dx;

if(abs(dx)>abs(dy))

l=(abs(dx));

else

l=(abs(dy));

glClear(GL\_COLOR\_BUFFER\_BIT);

glBegin(GL\_POINTS);

glVertex2d(x,y);

for(int i=0; i<l; i++)

{

if(ds>0)

{

int m=2\*(dy-dx);

x = x+1;

y = y+1;

glVertex2d(round(x), round(y));

ds=ds+m;

}

else if(ds<=0)

{

int m=2\*dy;

x=x+1;

y=y;

glVertex2d(round(x), round(y));

ds=ds+m;

}

}

glEnd();

glFlush();

}

void myInit (void)

{

glClearColor(0.0, 0.0, 0.0, 0.0);

glColor3f(1.0f, 1.0f, 0.0f);

glPointSize(4.0);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

gluOrtho2D(-100.0, 640.0,-100.0, 480.0);

}

int main(int argc, char\*\* argv)

{

glutInit(&argc, argv);

glutInitDisplayMode (GLUT\_SINGLE | GLUT\_RGB);

glutInitWindowSize (640, 480);

glutInitWindowPosition (100, 150);

glutCreateWindow ("DDA");

cout<<"Enter an initial points:\t"<<endl;

cin>>X1;

cin>>Y1;

cout<<"Enter the final points:\t"<<endl;

cin>>X2;

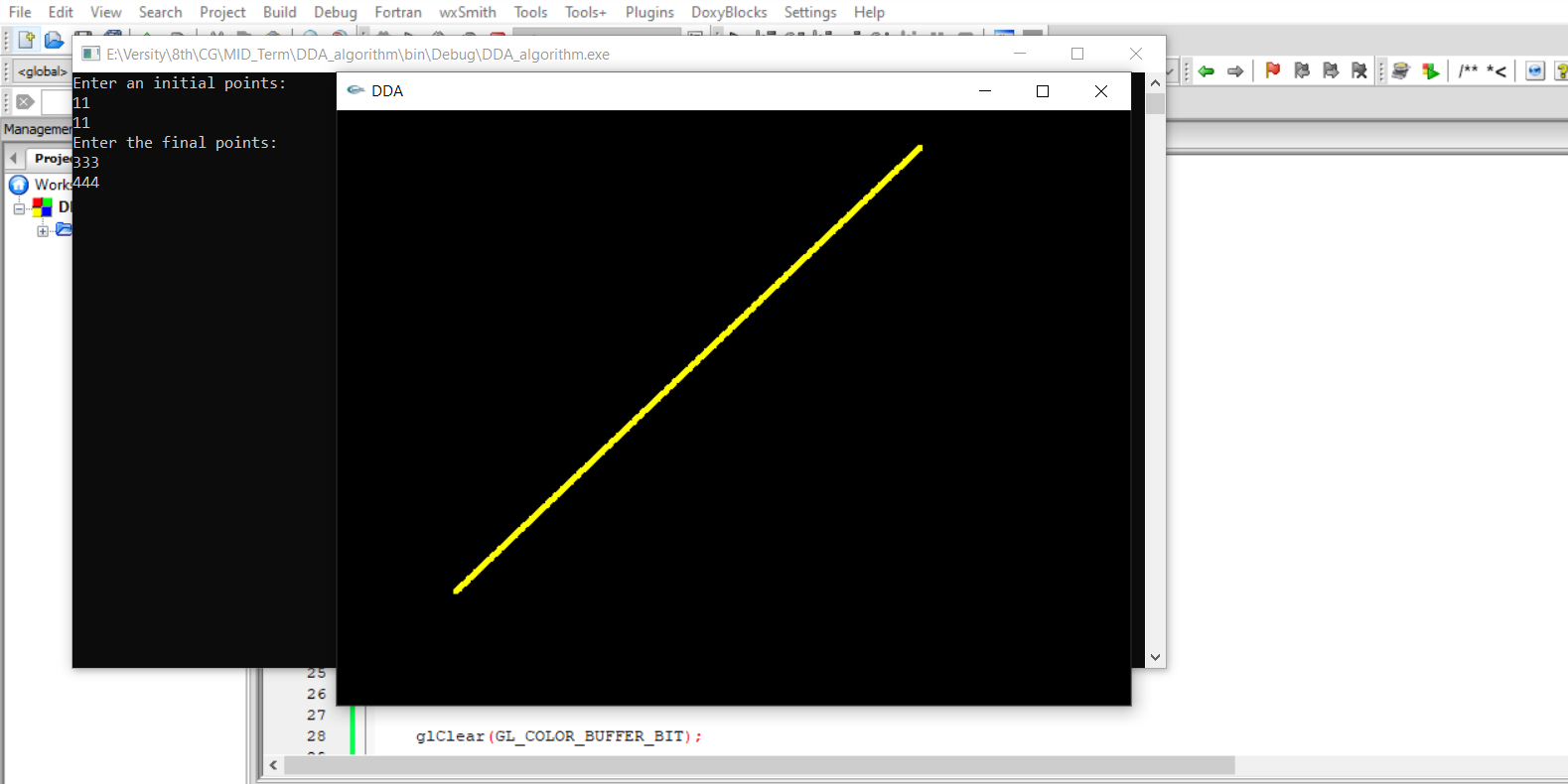
cin>>Y2;

glutDisplayFunc(DDA);

myInit ();

glutMainLoop();

}



**Answer Qusetion no:2**

#include <windows.h>

#include <GL/glut.h>

void display()

{

glClearColor(0.0f, 0.0f, 0.0f, 1.0f);

glClear(GL\_COLOR\_BUFFER\_BIT);

glBegin(GL\_TRIANGLE\_FAN);

glColor3f(1.0f,0.0f,0.6f);

glVertex2f(0.0f, 0.0f);

glVertex2f(0.5f, 0.0f);

glVertex2f(0.49f, 0.1f);

glVertex2f(0.45f, 0.2f);

glVertex2f(0.4f, 0.3f);

glVertex2f(0.3f, 0.4f);

glVertex2f(0.2f, 0.45f);

glVertex2f(0.1f, 0.49f);

glVertex2f(0.0f, 0.5f);

glVertex2f(-0.1f, 0.49f);

glVertex2f(-0.2f, 0.46f);

glVertex2f(-0.3f, 0.4f);

glVertex2f(-0.4f, 0.3f);

glVertex2f(-0.45f, 0.2f);

glVertex2f(-0.49f, 0.1f);

glVertex2f(-0.5f, 0.0f);

glVertex2f(-0.49f, -0.1f);

glVertex2f(-0.45f, -0.2f);

glVertex2f(-0.4f, -0.3f);

glVertex2f(-0.3f, -0.4f);

glVertex2f(-0.2f, -0.46f);

glVertex2f(-0.1f, -0.49f);

glVertex2f(-0.0f, -0.5f);

glVertex2f(0.1f, -0.49f);

glVertex2f(0.2f, -0.45f);

glVertex2f(0.3f, -0.4f);

glVertex2f(0.4f, -0.3f);

glVertex2f(0.45f, -0.2f);

glVertex2f(0.49f, -0.1f);

glVertex2f(0.5f, -0.0f);

glEnd();

glFlush(); // Render now

}

int main(int argc, char\*\* argv)

{

glutInit(&argc, argv);

glutCreateWindow("Triangle\_Fan");

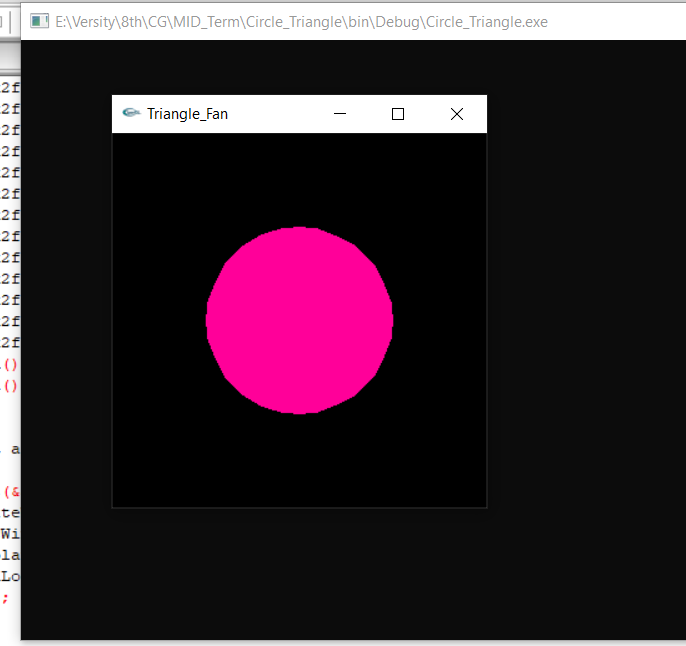
glutInitWindowSize(320, 320);

glutDisplayFunc(display);

glutMainLoop();

return 0;

}



**Answer Qusetion no:3**

#include <stdio.h>

#include <GL/gl.h>

#include <GL/glut.h>

float x=0,y,x2,y2,m,i,j,p;

int dx=0,dy=0,r;

void display(void)

{

glClear (GL\_COLOR\_BUFFER\_BIT);

glEnd();

glColor3f (0.6, 0.0, 1.0);

glBegin(GL\_POINTS);

p=1-r;

while((x<=y))

{

if(p<0)

{

x=x+1;

y=y;

printf("%0.2f %0.2f\n",x,y);

p=p+(2\*x)+1;

}

else

{

x=x+1;

y=y-1;

printf("%0.2f %0.2f\n",x,y);

p=p+(2\*x)+1-(2\*y);

}

glVertex3f (((x/100)), ((y/100)), 0.0);

glVertex3f (((y/100)), ((x/100)), 0.0);

glVertex3f ((-(x/100)), (-(y/100)), 0.0);

glVertex3f ((-(x/100)), ((y/100)), 0.0);

glVertex3f (((x/100)), (-(y/100)), 0.0);

glVertex3f (((y/100)), (-(x/100)), 0.0);

glVertex3f ((-(y/100)), (-(x/100)), 0.0);

glVertex3f ((-(y/100)), ((x/100)), 0.0);

}

glEnd();

glFlush ();

}

void init (void)

{

glClearColor (0.0, 0.0, 0.0, 0.0);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

glOrtho(-1.0, 1.0, -1.0, 1.0, -1.0, 1.0);

}

int main(int argc, char\*\* argv)

{

printf("Enter radius: ");

scanf("%d",&r);

y=r;

glutInit(&argc, argv);

glutInitDisplayMode (GLUT\_SINGLE | GLUT\_RGB);

glutInitWindowSize (500, 500);

glutInitWindowPosition (100, 100);

glutCreateWindow ("circle");

init ();

glutDisplayFunc(display);

glutMainLoop();

return 0;

}

