**Program 12:**

**Write a program to construct Bezier curve. Control points are supplied through keyboard/ mouse.**

#include<iostream>

#include<math.h>

#include<gl/glut.h>

using namespace std;

float f, g, r, x1[4], yc[4];

int flag = 0;

void myInit() {

glClearColor(1, 1, 1, 1);

glColor3f(1, 1, 1);

glPointSize(5);

gluOrtho2D(0, 500, 0, 500);

}

void drawPixel(float x, float y) {

glBegin(GL\_POINTS);

glVertex2f(x, y);

glEnd();

}

void display() {

glClear(GL\_COLOR\_BUFFER\_BIT);

int i;

double t;

glColor3f(0, 0, 0);

glBegin(GL\_POINTS);

for (t = 0; t < 1; t = t + 0.005) {

double xt = pow(1 - t, 3) \* x1[0] + 3 \* t \* pow(1 - t, 2) \* x1[1] + 3 \* pow(t, 2) \* (1 - t) \* x1[2] + pow(t, 3) \* x1[3];

double yt = pow(1 - t, 3) \* yc[0] + 3 \* t \* pow(1 - t, 2) \* yc[1] + 3 \* pow(t, 2) \* (1 - t) \* yc[2] + pow(t, 3) \* yc[3];

glVertex2f(xt, yt);

}

glColor3f(1, 1, 0);

for (i = 0; i < 4; i++) {

glVertex2f(x1[i], yc[i]);

glEnd();

glFlush();

}

}

void mymouse(int btn, int state, int x, int y)

{

if (btn == GLUT\_LEFT\_BUTTON && state == GLUT\_DOWN && flag < 4)

{

x1[flag] = x;

yc[flag] = 500 - y;

cout << " X: " << x << " Y" << 500 - y;

glPointSize(3);

glColor3f(1, 1, 0);

glBegin(GL\_POINTS);

glVertex2i(x, 500 - y);

glEnd();

glFlush();

flag++;

}

if (flag >= 4 && btn == GLUT\_LEFT\_BUTTON)

{

glColor3f(0, 0, 1);

display();

flag = 0;

}

}

int main(int argc, char\* argv[]) {

glutInit(&argc, argv);

/\*

//USE KEYBOARD

cout << "Enter the x co-ordinates";

cin >> x1[0] >> x1[1] >> x1[2] >> x1[3];

cout << "Enter y co-ordinates";

cin >> yc[0] >> yc[1] >> yc[2] >> yc[3];

//END KEYBOARD

\*/

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

glutInitWindowSize(500, 500);

glutInitWindowPosition(0, 0);

glutCreateWindow("BZ");

glutDisplayFunc(display);

glutMouseFunc(mymouse); //INCLUDE FOR MOUSE, REMOVE FOR KEYBOARD

myInit();

glutMainLoop();

}

**OUTPUT:**



