Name : Shawn Louis	Batch : B	Roll No : 31

EXPERIMENT 11

Title

IMPLEMENTATION OF BEZIER CURVE

Objective

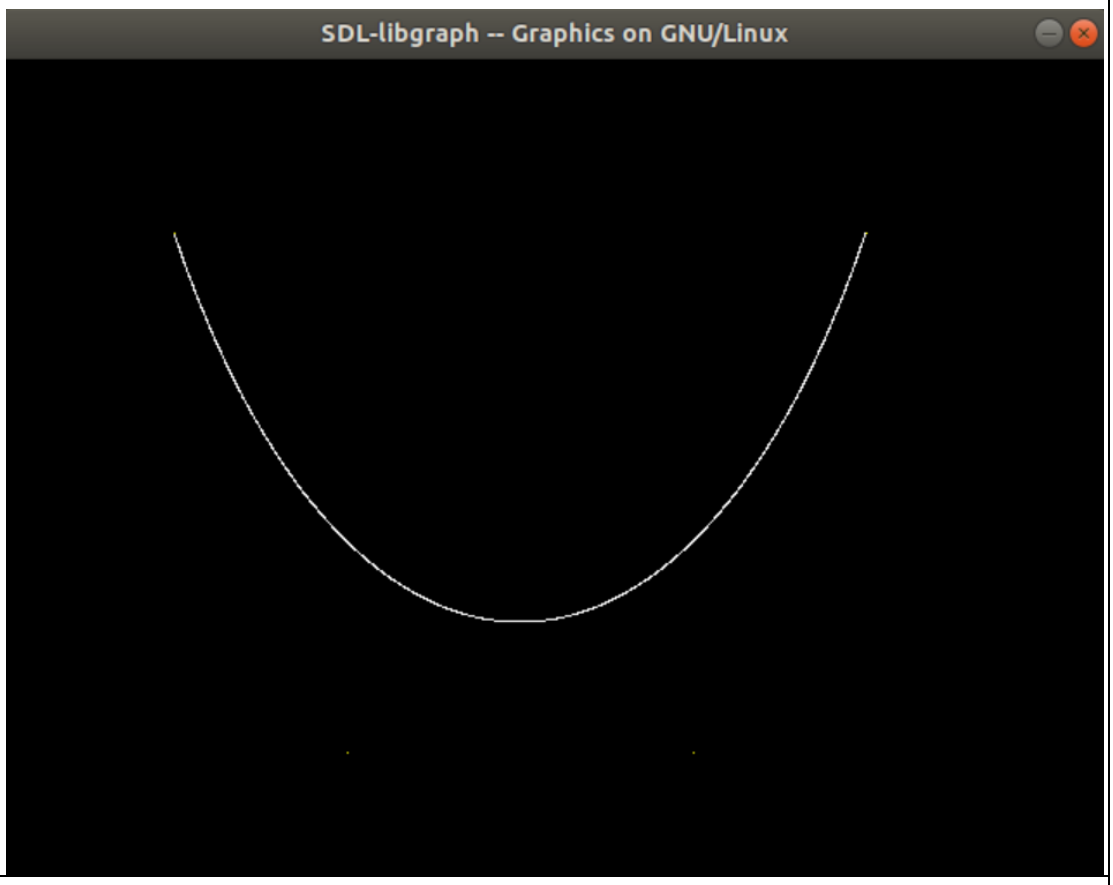
To write a C program for Bezier Curve.

Program

```
#include <stdio.h>
#include <stdlib.h>
#include <graphics.h>
#include <math.h>
void bezier (int x[4], int y[4])
{
int gd = DETECT, gm;
int i;
double t;
initgraph (&gd, &gm, "C:\\TurboC3\\BGI");
for (t = 0.0; t < 1.0; t += 0.0005)
{
double xt = pow (1-t, 3) * x[0] + 3 * t * pow (1-t, 2) * x[1] +3 *
pow (t, 2) * (1-t) * x[2] + pow (t, 3) * x[3];
double yt = pow (1-t, 3) * y[0] + 3 * t * pow (1-t, 2) * y[1] +3 *
pow (t, 2) * (1-t) * y[2] + pow (t, 3) * y[3];
putpixel (xt, yt, WHITE);
}
for (i=0; i<4; i++)
putpixel (x[i], y[i], YELLOW);
getch();
closegraph();
return;
}
void main()
{
int x[4], y[4];
int i;
printf ("Enter the x- and y-coordinates of the four control points.\n");
for (i=0; i<4; i++)
scanf ("%d%d", &x[i], &y[i]);
bezier (x, y);
}
```

Output

```
shawn@shawn-VirtualBox:~/Desktop$ gedit BezierCurve.c
shawn@shawn-VirtualBox:~/Desktop$ gcc BezierCurve.c -lgraph -lm -ldl
shawn@shawn-VirtualBox:~/Desktop$ ./a.out
Enter the x- and y-coordinates of the four control points.
100 100
200 300
300 300
400 100
```

	
Conclusion	Thus a C program to generate Bezier Curve was written and executed.