**Practical No: 13**

**Aim :** Write a C Program to implement the B-Spline Curve.

**Performed By :** Piyush Vanjari

**Class :** BCA-III SEM-V

**Date :** 10/09/2024

**Code :**

#include <stdio.h>

#include <graphics.h>

#include <math.h>

#define MAX\_POINTS 10

#define MAX\_SEGMENTS (MAX\_POINTS - 3)

#define STEPS 100

int points[MAX\_POINTS][2] = {

{100, 300}, {150, 100}, {300, 50}, {400, 200},

{500, 300}, {500, 400}, {700, 500}, {800, 400},

{890, 300}, {950, 200}

};

float basis(int i, int k, float t, float knots[]) {

if (k == 1) {

if (knots[i] <= t && t < knots[i + 1])

return 1.0;

return 0.0;

}

float denom1 = knots[i + k - 1] - knots[i];

float denom2 = knots[i + k] - knots[i + 1];

float term1 = 0.0, term2 = 0.0;

if (denom1 != 0)

term1 = ((t - knots[i]) / denom1) \* basis(i, k - 1, t, knots);

if (denom2 != 0)

term2 = ((knots[i + k] - t) / denom2) \* basis(i + 1, k - 1, t, knots);

return term1 + term2;

}

void drawBSpline(int n, int k, float knots[]) {

int gd = DETECT, gm;

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

for (int i = 0; i < STEPS; i++) {

float t = knots[k - 1] + (knots[n + 1] - knots[k - 1]) \* i / (float)STEPS;

float x = 0.0, y = 0.0;

for (int j = 0; j <= n; j++) {

float b = basis(j, k, t, knots);

x += b \* points[j][0];

y += b \* points[j][1];

}

putpixel((int)x, (int)y, WHITE);

}

getch();

closegraph();

}

int main() {

int n = 9; // Number of control points - 1

int k = 4; // Order of the B-spline

float knots[MAX\_POINTS + k] = {0, 0, 0, 0, 1, 2, 3, 4, 5, 6, 6, 6, 6};

drawBSpline(n, k, knots);

return 0;

}

**Output :**

