

## fittingPolynomial.c

```
1  #include <stdio.h>
2  #include <math.h>
3  #define MAX 10
4
5  void normal(float x[MAX], float y[MAX], float c[MAX][MAX], float b[MAX], int n, int m)
6  {
7      int l1, l2;
8
9      for (int j = 0; j < m; j++)
10     {
11         for (int k = 0; k < m; k++)
12         {
13             c[j][k] = 0.0;
14             l1 = k + j - 1;
15             for (int i = 0; i < n; i++)
16             {
17                 c[j][k] = c[j][k] + pow(x[i], l1);
18             }
19         }
20     }
21     for (int j = 0; j < m; j++)
22     {
23         b[j] = 0.0;
24         l2 = j - 1;
25
26         for (int i = 0; i < n; i++)
27         {
28             b[j] = b[j] + y[i] * pow(x[i], l2);
29         }
30     }
31     return;
32 }
33
34 void gauss(int n, float a[MAX][MAX], float b[MAX], float x[MAX])
35 {
36     // int i, j, k;
37     float pivot, factor, sum;
38     for (int k = 0; k < n - 1; k++)
39     {
40         pivot = a[k][k];
41         for (int i = k + 1; i < n; i++)
42         {
43             factor = a[i][k] / pivot;
44             for (int j = k + 1; j < n; j++)
45             {
46                 a[i][j] = a[i][j] - factor * a[k][j];
47             }
48             b[i] -= factor * b[k];
49         }
50     }
51
52     x[n] = b[n] / a[n][n];
```

```
53     for (int k = n - 1; k >= 0; k--)
54     {
55         sum = 0.0;
56         for (int j = k + 1; j < n; j++)
57         {
58             sum = sum + a[k][j] * x[j];
59         }
60         x[k] = (b[k] - sum) / a[k][k];
61     }
62     return;
63 }
64
65 int main()
66 {
67     int n, mp, m;
68     float x[MAX], y[MAX], c[MAX][MAX], a[MAX], b[MAX];
69
70     /* Reading Values */
71     printf("\nInput number of data points: ");
72     scanf("%d", &n);
73
74     printf("\nInput degree of required polynomial: ");
75     scanf("%d", &mp);
76
77     printf("\nInput x and y values (one set on each line): ");
78     for (int i = 0; i < n; i++)
79     {
80         scanf("%f %f", &x[i], &y[i]);
81     }
82
83     /* Number of polynomial coefficients */
84     m = mp + 1;
85
86     /* Computation of elements of c and b */
87     normal(x, y, c, b, n, m);
88
89     /* Computation of coefficients a(1) to a(m) */
90     gauss(m, c, b, a);
91
92     /* Printing of coefficients a(i) */
93     printf("\nPolynomial Coefficients\n\n");
94     for (int i = 0; i < m; i++)
95     {
96         printf("%15.6f", a[i]);
97     }
98     return 0;
99 }
100
```