

~\OneDrive\Desktop\Newton divided.c\tempCodeRunnerFile.c

```
1 #include <stdio.h>
2
3 // Function to calculate factorial
4 int factorial(int n) {
5     if (n == 0)
6         return 1;
7     else
8         return n * factorial(n - 1);
9 }
10
11 // Function to calculate the forward difference table
12 void forwardDifference(double f[], int n, double diff[][]) {
13     for (int i = 0; i < n; i++)
14         diff[i][0] = f[i];
15     for (int j = 1; j < n; j++) {
16         for (int i = 0; i < n - j; i++) {
17             diff[i][j] = diff[i + 1][j - 1] - diff[i][j - 1];
18         }
19     }
20 }
21
22 // Function to calculate Newton's forward difference polynomial
23 double newtonsForwardDifference(double x, double xi[], double f[], int n, double diff[][]) {
24     double result = f[0];
25     double term = 1;
26     double u = (x - xi[0]) / (xi[1] - xi[0]);
27     for (int i = 1; i < n; i++) {
28         term *= u - i + 1;
29         result += (term * diff[0][i]) / factorial(i);
30     }
31     return result;
32 }
33
34 int main() {
35     int n;
36     printf("Enter the number of data points: ");
37     scanf("%d", &n);
38
39     double xi[n], f[n];
40     printf("Enter the data points (x, f(x)):\n");
41     for (int i = 0; i < n; i++) {
42         scanf("%lf %lf", &xi[i], &f[i]);
43     }
44
45     double diff[n][n];
46     forwardDifference(f, n, diff);
47
48     double x;
49     printf("Enter the value of x for interpolation: ");
50     scanf("%lf", &x);
51
52     double result = newtonsForwardDifference(x, xi, f, n, diff);
```

```
53     printf("Interpolated value at x = %.2lf is %.2lf\n", x, result);
54
55     return 0;
56 }
```