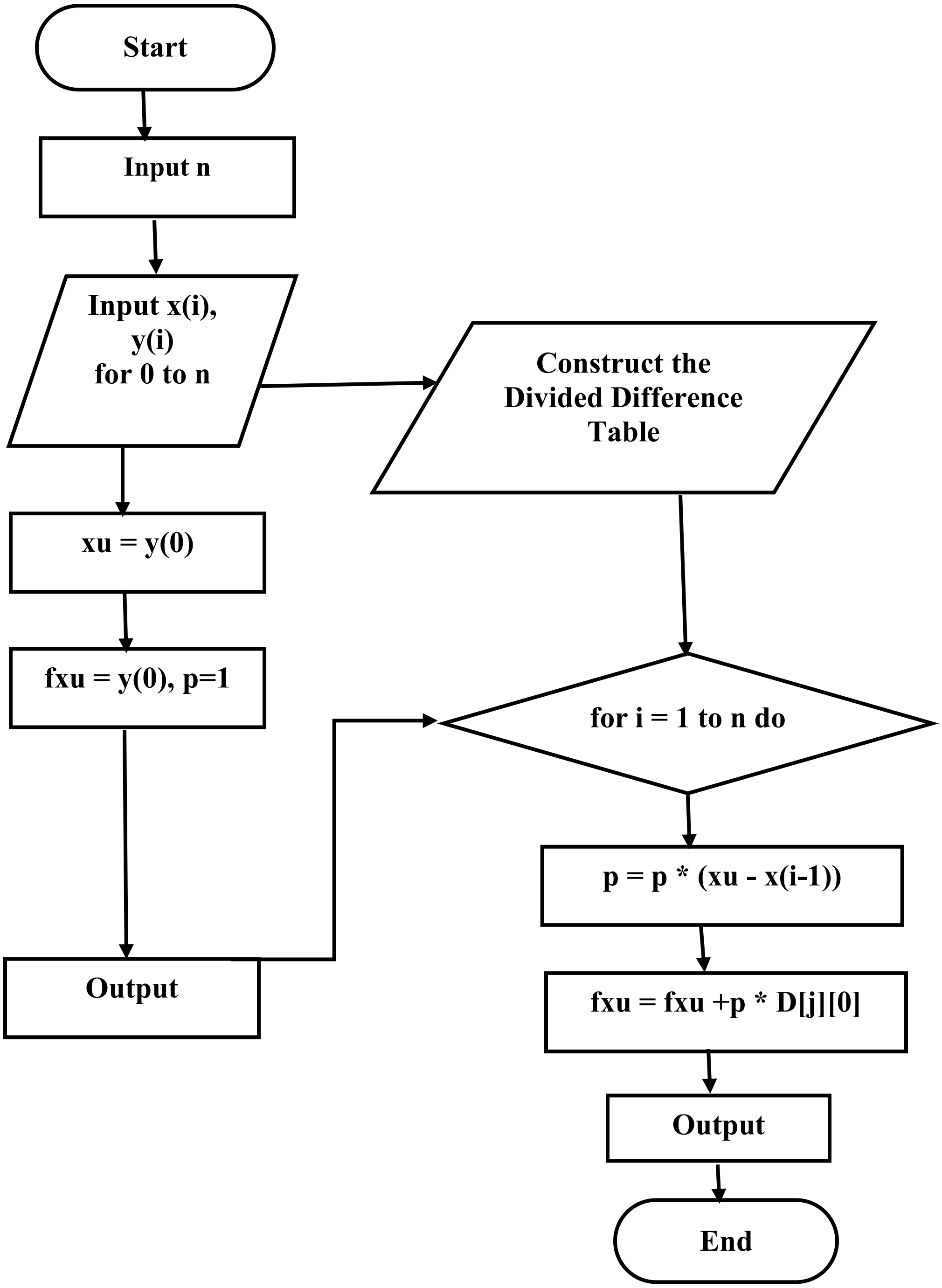
1.Flow Chart for the Newton Divided Difference Method



2. Newton Divided Difference

```
public class NewtonDividedDifference {

    // Function to calculate divided differences
    public static double[][] dividedDifferenceTable(double[] x, double[] y) {
        int n = x.length;
        double[][] table = new double[n][n];

        // First column is y[]
        for (int i = 0; i < n; i++) {
            table[i][0] = y[i];
        }

        // Calculating divided difference table
        for (int j = 1; j < n; j++) {
            for (int i = 0; i < n - j; i++) {
                table[i][j] = (table[i + 1][j - 1] - table[i][j - 1]) / (x[i + j] - x[i]);
            }
        }

        return table;
    }

    // Function to get the interpolation polynomial value at a given point
    public static double newtonInterpolation(double[][] table, double[] x, double value) {
        int n = x.length;
        double result = table[0][0];
        double term;
        for (int i = 1; i < n; i++) {
            term = table[0][i];
            for (int j = 0; j < i; j++) {
                term *= (value - x[j]);
            }
            result += term;
        }

        return result;
    }

    public static void main(String[] args) {
        // Sample data points
        double[] x = {5, 6, 9, 11};
        double[] y = {12, 13, 14, 16};

        double[][] table = dividedDifferenceTable(x, y);

        System.out.println("Divided Difference Table:");
        for (int i = 0; i < x.length; i++) {
            for (int j = 0; j <= i; j++) {
                System.out.printf("%10.4f ", table[i - j][j]);
            }
            System.out.println();
        }

        double value = 7;
        double result = newtonInterpolation(table, x, value);
        System.out.printf("Interpolated value at x = %.2f is %.4f\n", value, result);
    } }
```