Transcendental Regression Program Documentation

Objective

The objective of this program is to implement three types of transcendental regression models—exponential, logarithmic, and power-law—to model the relationship between two variables, x and y. The program computes the regression function and predicts the value of y for a given input x.

Concept

Transcendental regression is used to model data that does not fit well into a simple linear relationship. This program implements three common models:

- **Exponential regression:** y = a * e^(b * x), used when growth/decay is exponential.
- **Logarithmic regression:** y = a + b * ln(x), used when growth slows as x increases.
- **Power-law regression:** y = a * x^b, used when y varies as a power of x.

The regression coefficients a and b are calculated using least squares fitting of transformed variables, so that the problem can be solved using linear regression techniques on the transformed data.

Program Features

- Reads input data from a file (input.txt) and writes output to a file (output.txt).
- Handles multiple test cases.
- Computes regression coefficients a and b for exponential, logarithmic, and power-law regression.
- Predicts y for a given x input.
- Prints the regression function and predicted value to both console and file.
- Ignores invalid regression types silently.

Input Format

File: input.txt

- The first line contains the number of test cases.
- For each test case:
    - A line containing type (1 for exponential, 2 for logarithmic, 3 for power-law) and n (number of data points).
    - n lines, each containing x and y values separated by space.
    - A single value xInput for which prediction is required.

Example:

3

1 5

1 2

2 4

3 8

4 16

5 32

6

2 4

1 2

2 3

3 4

4 5

5

3 5

1 1

2 8

3 27

4 64

5 125

6

Output Format

File: output.txt (also printed on console)

For each test case:

Test Case 1

Exponential Regression

Function: y = a * e^(b * x)

Predicted y for x = X: Y

Example output:

Test Case 1

Exponential Regression

Function: y = 1.0000 * e^(0.6931 * x)

Predicted y for x=6: 64.0000

Formulas for Regression Coefficients

Exponential regression $y = ae^{bx}$

- Transform y to $Y = \ln(y)$
- $b = \frac{n\sum(x \cdot \ln(y)) - (\sum x)(\sum \ln(y))}{n\sum x^2 - (\sum x)^2}$
- $a = \exp\left(\frac{\sum \ln(y) - b\sum x}{n}\right)$

Logarithmic regression $y = a + b\ln(x)$

- **Transform x to $L = \ln(x)$**
- $b = \frac{n\sum(L \cdot y) - (\sum L)(\sum y)}{n\sum L^2 - (\sum L)^2}$
- $a = \frac{\sum y - b\sum L}{n}$

Power-law regression $y = ax^b$

- **Transform x and y: $LX = \ln(x), LY = \ln(y)$**
- $b = \frac{n\sum(LX \cdot LY) - (\sum LX)(\sum LY)}{n\sum LX^2 - (\sum LX)^2}$
- $a = \exp\left(\frac{\sum LY - b\sum LX}{n}\right)$

Algorithm

- Read number of test cases from input.
- For each test case:
    - Read regression type (1 = exponential, 2 = logarithmic, 3 = power-law) and number of data points n.
    - Read n data points (x, y).
    - Read xInput for prediction.
    - Depending on the type, compute regression coefficients using the formulas above.
    - Compute predicted y using the regression function.
    - Print and write the regression function and predicted y to console and file.

Features of This Program

- Supports multiple regression types in a single run.
- Computes regression coefficients using least squares method on transformed data.
- Predicts y for any given x input.

- Outputs are printed to console and saved in an output file.
- Handles multiple test cases sequentially.
- Ignores invalid regression types gracefully without errors.
- Can be extended for additional transcendental regression types if needed.