Experiment 01 Report

**Lokesh R M**

**BU22CSEN0100145**

# **Experiment Statement**

Implement weather modeling\* using the quadratic solution in stages:

* 1. hard-coding variables
  2. keyboard input
  3. read from a file for a single set of inputs & multiple sets of inputs.

Save all versions, debug, fix problems, and create a GitHub account.

# Focus of the Report

This report focuses on implementing and evaluating the hard-coded approach for solving the quadratic equation. The **Read from file (Single Set and Multiple Sets of Input)** version calculates the roots using fixed coefficients ***a***, ***b***, and ***c***.

# The Solution Approach

Temperature during the day often follows a pattern: it’s cooler in the morning, warms up by midday, and cools down again in the evening. This rise and fall can resemble a curve, which can be modeled using a quadratic equation:

***T(x)=ax2+bx+c***

Here:

* ***T(x)***: Predicted temperature.
* ***x***: Time (e.g., hours of the day).
* ***a,b,c***: Coefficients calculated from historical data.

Given data:

* 6 AM (x = 6): 15°C
* 12 PM (x = 12): 25°C
* 6 PM (x = 18): 20°C

The resulting equation:

T(x) = -0.5x2 + 12x - 40

Prediction for 3 PM (x = 15):  
T(15) = -0.5(15)2 + 12(15) - 40 = 22.5°C

# Code Implementation for the Version (V3)

Read from a file for a single set of inputs & multiple sets of inputs.

## 4.1 Read from a File (Single Set of Inputs)

### 4.1.1 Observation

The program can read math problem details from a file and solve quadratic equations well. It handles different situations, like when no real answers exist. The solution works correctly, giving the right answers for test cases. By using a file to input numbers, the program makes math calculations easier and more accurate.

|  |  |
| --- | --- |
| Parameters | Single File Input |
| Execution Time | < 1s |
| Error Handling | Handled file and data format errors effectively |
| Scalability | N/A |
| User Experience | Easy to prepare input files  Clear Error Message |

### 4.1.2 Challenges Faced

We carefully validated the file input format and ensured our program could handle various challenges. We tested scenarios like unopenable files or invalid data, paying close attention to precise math calculations. We created a robust and reliable quadratic equation solver by managing complex edge cases and potential errors.

### 4.1.3 Further Updates

We expanded our program to process multiple quadratic equations from a single file, enabling efficient batch calculations. We strengthened error handling to manage complex scenarios like division by zero and added data validation to prevent unexpected issues. We made our solution more comprehensive and informative by implementing user-friendly error messages and incorporating curve visualization.

## 4.2 Read from a File (Multiple Set of Inputs)

### 4.2.1 Observation

The program efficiently processes multiple sets of inputs from a single file, solving each quadratic equation and displaying the results consecutively. This batch-processing capability significantly improves scalability, making it suitable for handling large datasets. The program performs as expected, accurately computing the roots for all input cases and identifying edge cases such as equations with no real roots.

|  |  |
| --- | --- |
| Parameters | Multiple File Input |
| Execution Time | <1s per dataset |
| Error Handling | Robust error handling for file and data issues |
| Scalability | Handles large datasets effectively |
| User Experience | Easy to format input files |
| Clear Error Message | Yes |

### 4.2.2 Challenges Faced

* Designing a format for the input file that can accommodate multiple quadratic equations while remaining easy to prepare and understand.
* Implementing a loop to read and process each set of coefficients without skipping or duplicating any data.
* Managing edge cases such as missing coefficients, invalid input values, or excessively large datasets that could strain memory or processing power.

### 4.2.3 Further Updates

* Enhanced the user interface to provide summary results after processing all inputs, including the total number of equations solved, the number with real roots, and those without.
* Optimized file reading logic to improve performance when handling very large datasets.
* Introduced detailed error reporting to identify and log problematic input cases for easier debugging and correction.

## Conclusion

Implementing single and multiple input processing versions demonstrates the program’s flexibility and reliability in solving quadratic equations. By automating data input from files, the program reduces manual errors, enhances scalability, and ensures consistent results.

The ability to handle multiple sets of inputs efficiently positions the program as a robust tool for applications requiring batch processing of mathematical problems. Future improvements can enhance its utility, such as adding graphical output or integrating larger data analysis pipelines.