**PHYSICS CALCULATOR**

**A python program presented to Mr. Allan O. Ibo Jr.**

**in fulfillment of the subject**

**CCCS 102 – Fundamentals of Programming**

**GUILLER ANGELO HERMOSO**

**JULIET PONTANARES**

**FERNANNE HANNAH A. ENIMEDEZ**

PROSPECTS

**PHYSICS CALCULATOR**

1. INTRODUCTION

The **Physics Calculator** is a python program that calculates for various physical quantitative measures.

The program aims to provide its user/s with a 20-in-1 calculator that can analyze a text file of Physics word problems and creates a new text file containing its answers.

In addition, users also have the option of manual input.

Physical quantities available for calculation include:

* Acceleration
* Buoyant Force
* Centripetal Acceleration
* Centripetal Force
* Displacement
* Elastic Potential Energy
* Force
* Friction Force
* Gravitational Force
* Gravitational Potential Energy
* Impulse
* Kinetic Energy
* Momentum
* Power
* Pressure
* Tension Force
* Velocity
* Weight
* Work
* Work done by Elastic Potential Energy

1. CODE SNIPPETS

Tasks 1 to 4 were accomplished within the Physics Calculator as all conditions under said tasks were met.

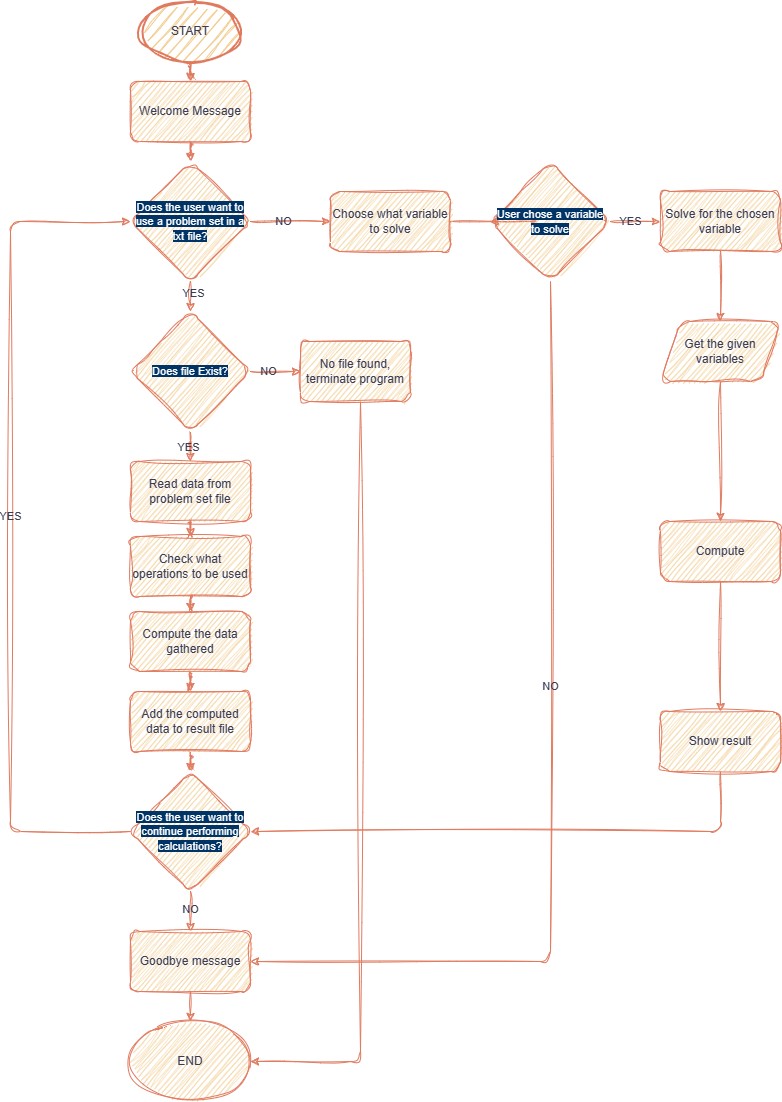
**TASK 1: Algorithm Design and Python Basics**

The Physical Calculator’s friendly input functionalities ask users to prompt input for the following data:

* Preferred mode of problem set input
  + Option 1: Use problem set from text file
  + Option 2: Manually input numerical values
* Physics operation to be performed
  + List of available operations are enumerated above.
* Appropriate values needed in solving for a particular physical quantity.

For the output, answers for each operation are either written in a new text file (for those who input problem set from text file) or displayed on the terminal (for those with manual input).

**ALGORITHM DESIGN: FLOWCHART**



**TASK 2: Operators and Control Structures**

The programmers nested if, for and while loop statements to perform the following commands:

* Read necessary variables from a text file containing word problems.
* Extract necessary values from lists and individual strings in lines.
* Condition the program to take input and perform operations in lieu of arguments given a specific set of parameters.
* Loop commands for invalid inputs to ensure correct input.
* Handle exceptions in case of unseen variable errors.

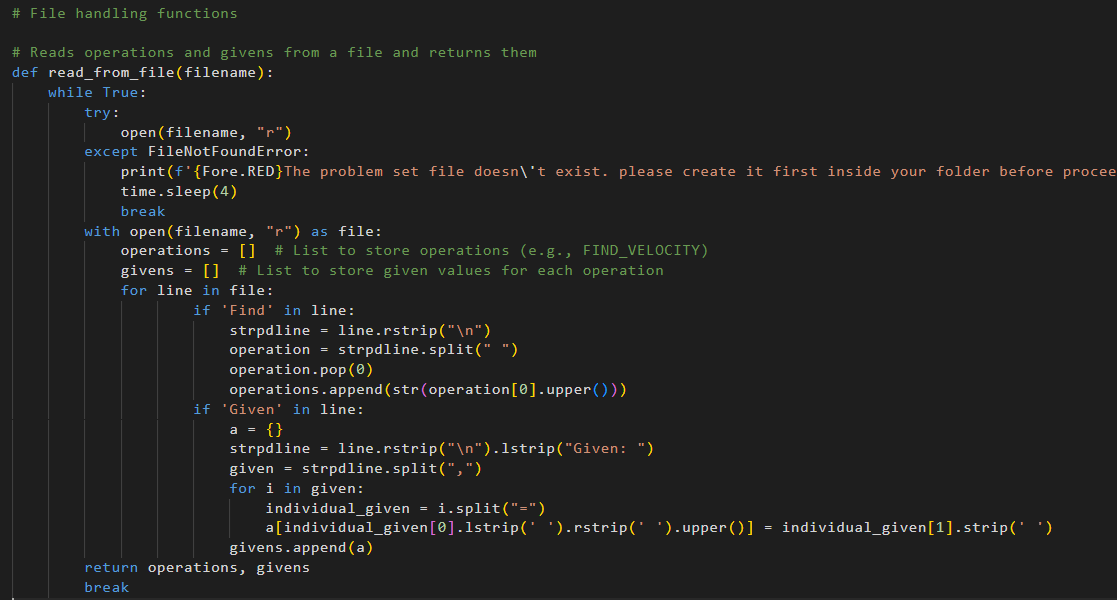
**TASK 3: String Manipulation & TASK 4: File Handling**

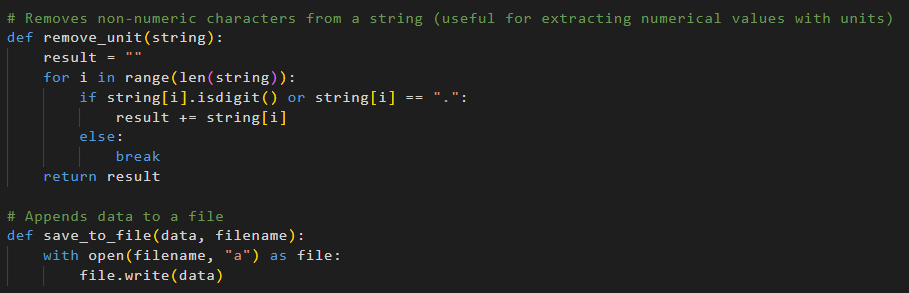
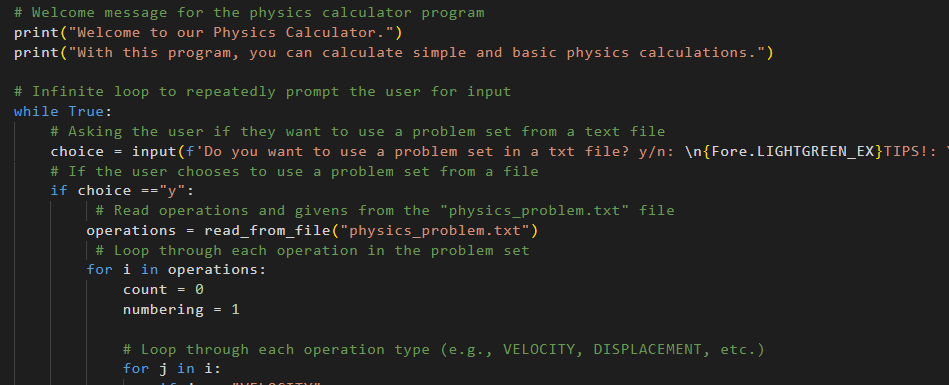
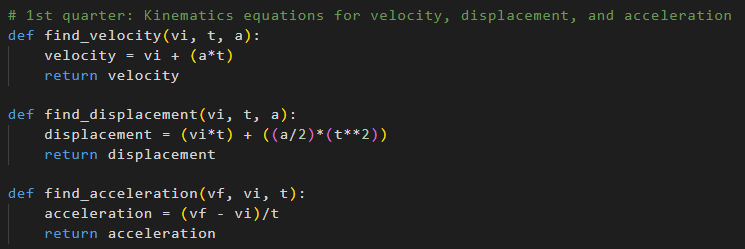
The string manipulation methods were applied for the option of using a file with problem set.

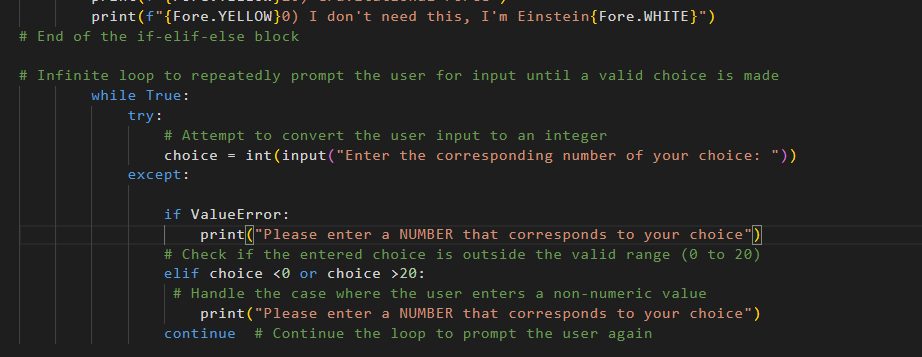
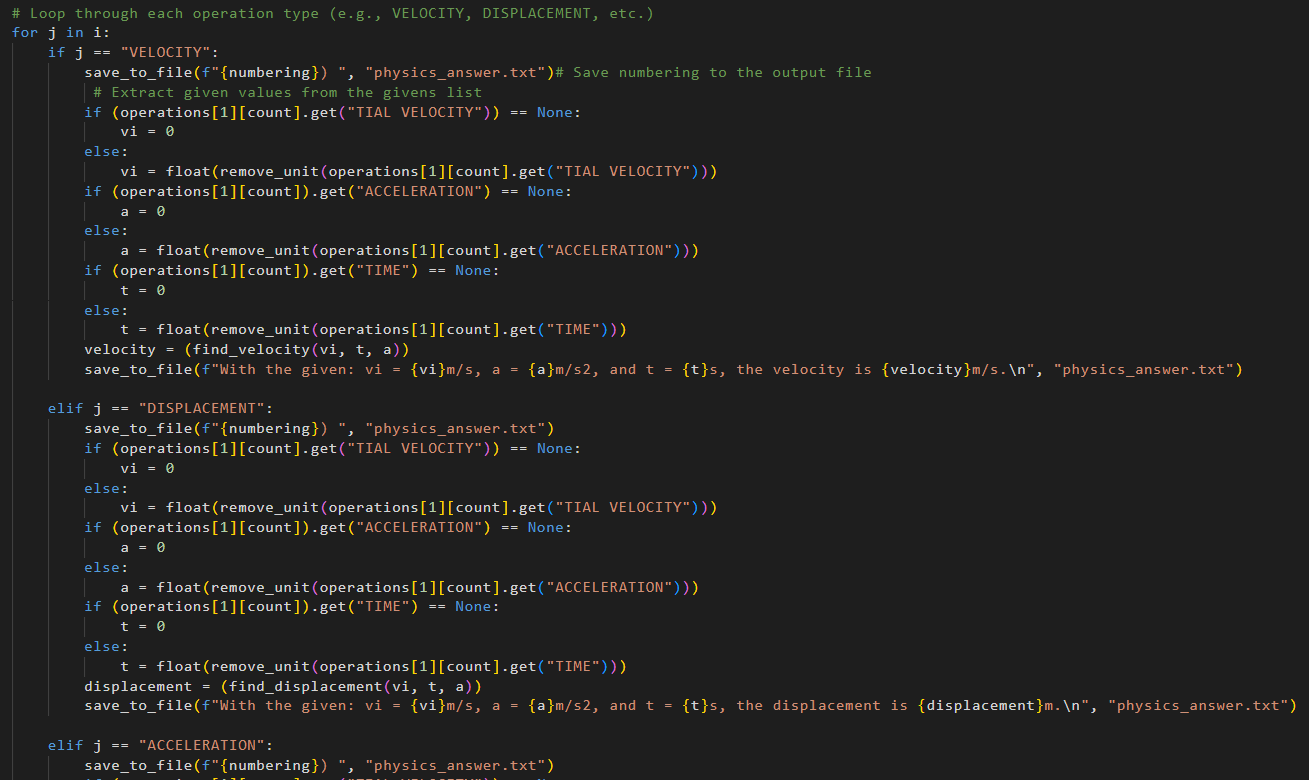
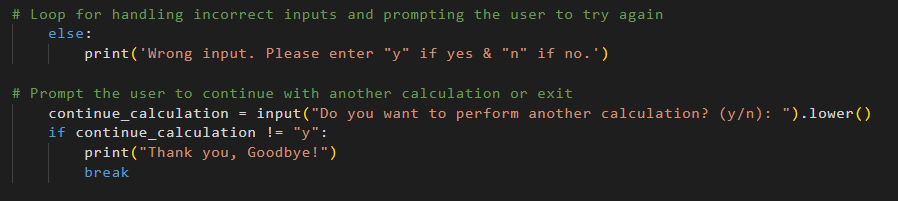
Because a text file is expected to contain alphanumeric characters, string manipulations under iterations help find and remove alphabets as well as so that it will only be able register numeric inputs into the program.

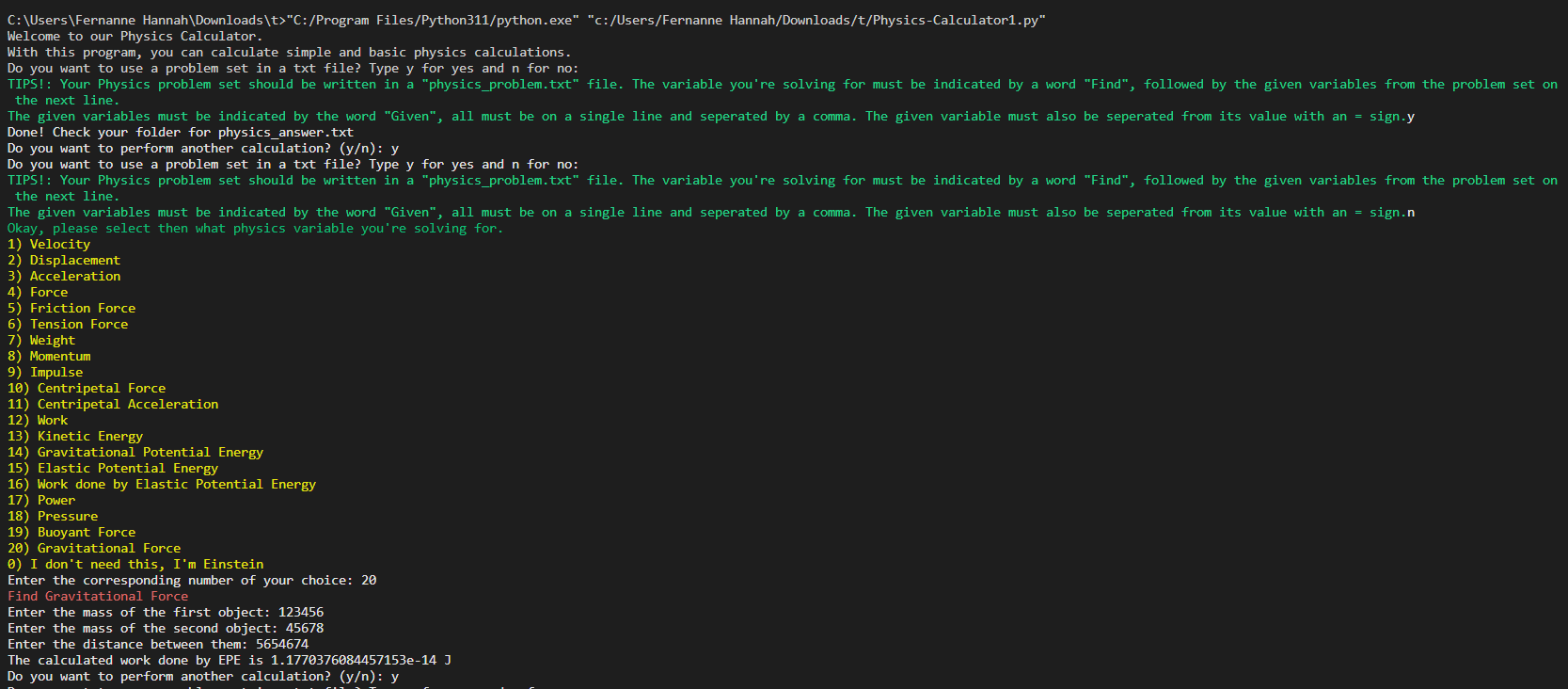
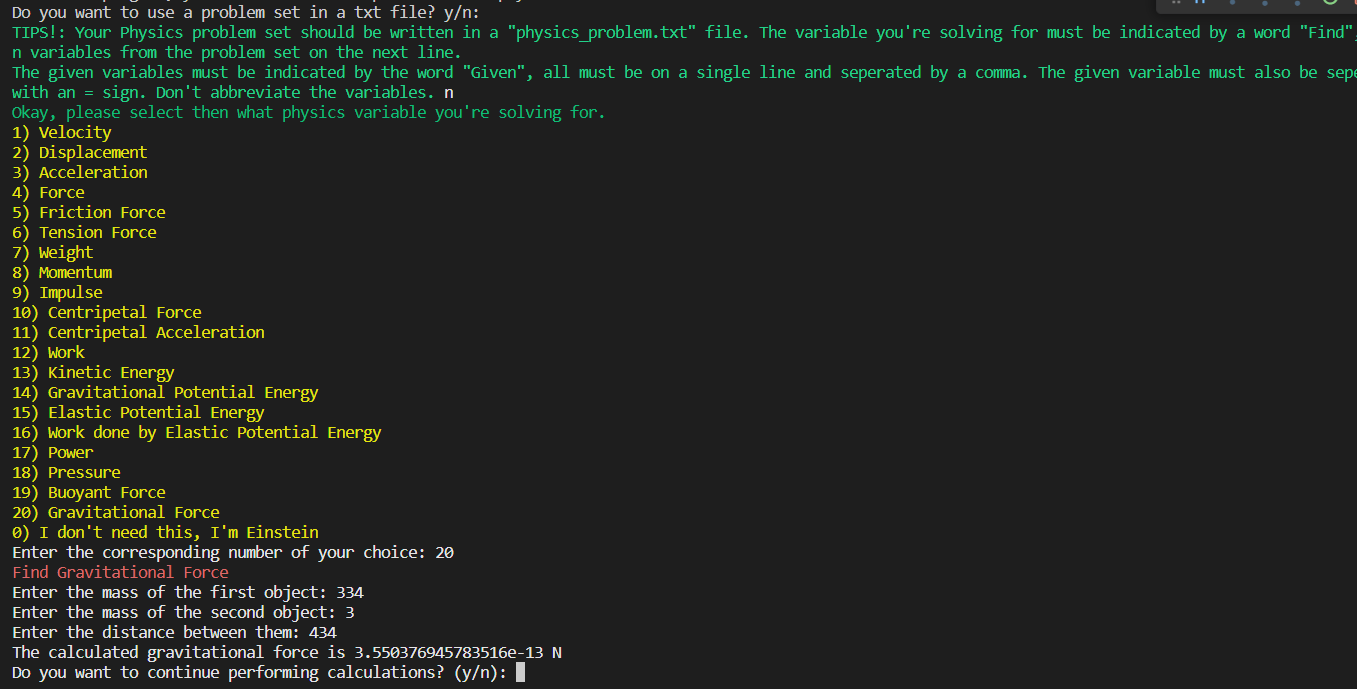
Meanwhile, file handling methods were used to link files into the program and access its data for analyzation, extraction, calculations, and string manipulation. It was also used in creating an external display site of results.

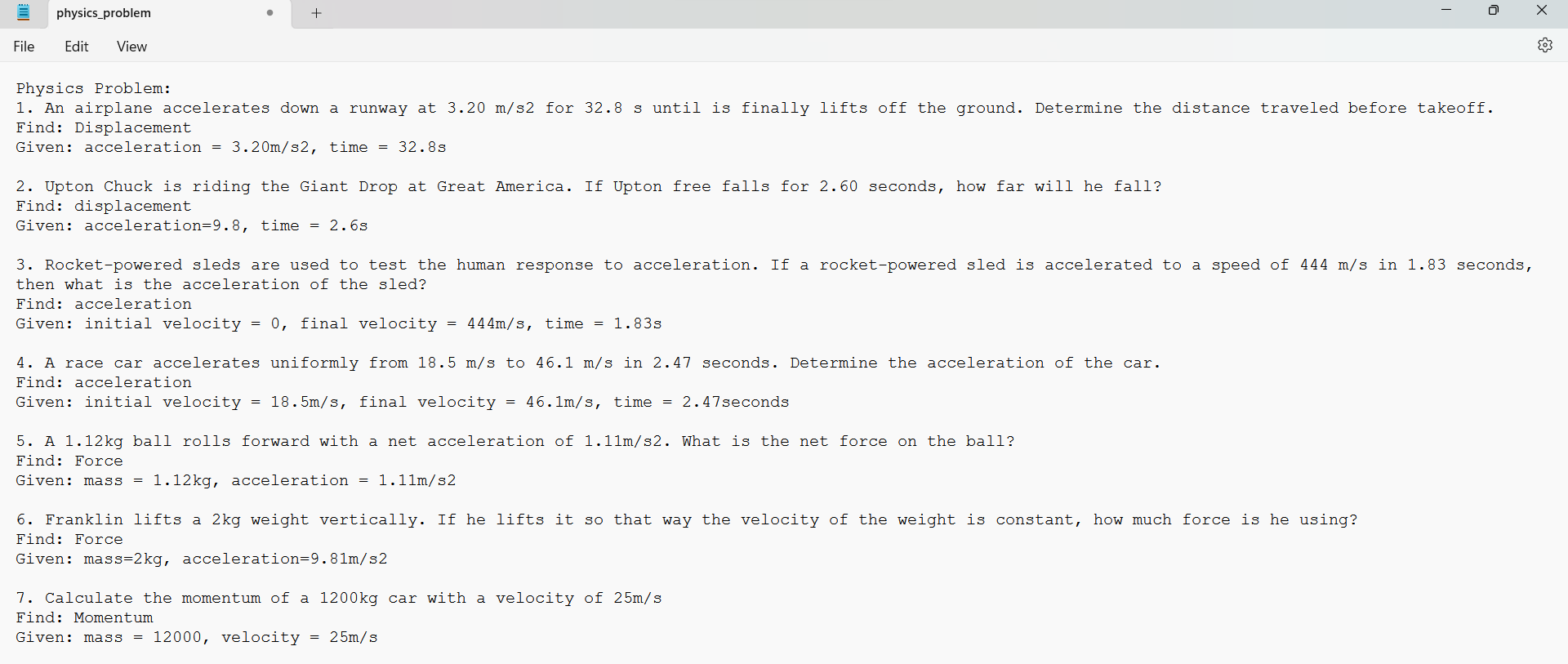
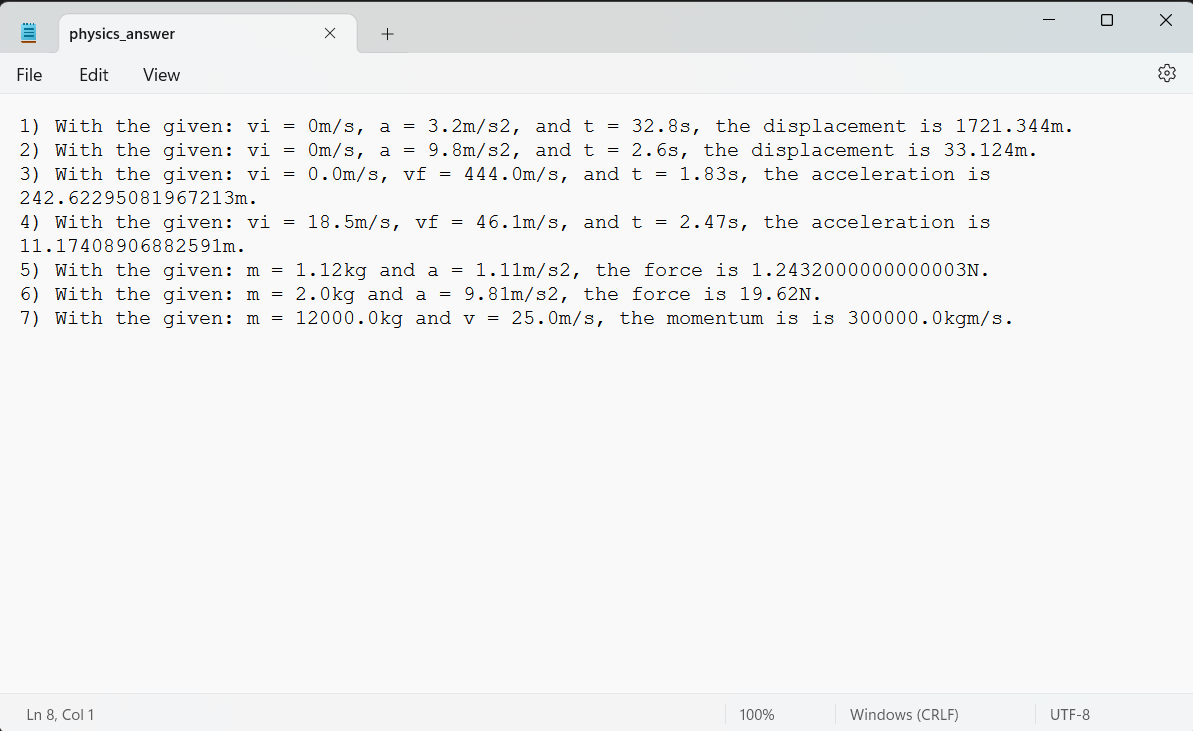
Below are the code snippets from the Physics Calculator with comments.







1. OUTPUT



1. CHALLENGES AND RESOLUTION

**Challenge 1: Reading the data from the file**

Reading and taking the data from the problem set file was a slightly complicated task since the program is oblivious about what the user have entered in the file.

Resolution

We added the tips that shows before the user use the file handling feature so that the user knows what should be the format of the problem set file before proceeding. We also used the if-loop to check every line of the text file for “Find” and “Given” keywords, that indicates the variable to be solved and given variables.

**Challenge 2: Matching the operation and variables**

After the program checks for keywords, it stores what it finds to a list and dictionary, that is both nested in a tuple.

Resolution

With that situation, we used if-loop to check for the operation keyword inside the list in tuple. Then to match the given variables to their respective operation, we used “operations[1][count]” to get what dictionary must be used. The count variable was set before the loop started so that is value won’t be reset.

1. CONCLUSION AND KEY TAKEAWAYS

The creation of the Physics Calculator was possible using Python’s functions, concepts, and libraries for user interaction: input-output, fundamental operations, logical arguments, control structures, string manipulation, and file and exception handling methods.

During the creation of this program, the key takeaways have been:

* + Algorithm first. Code after.
  + Research is a must in coding.
  + There are varying solutions to one problem – always opt for the efficient ones.
  + Insert comments to mark and explain the gist of the purpose of a block of code within the code.