Project: Word Frequency Counter Using Hash Tables

Due date: Friday December 5

In this project, you will analyze a text file and build a simple word-frequency counter. The goal is to practice using hashing strategies and hash-table–based data structures (like dictionaries) for efficient lookups.

Dataset

Download the dataset **adventures_of_huckleberry_finn.txt** by Mark Twain.

Main Task

Create a Python program that:

1. Reads the text file and extracts words.
2. Counts how many times each unique word appears.
3. Stores each word and its frequency using a **hash table** (dictionary).
4. Outputs the **20 most frequent words** in a table format.

Requirements

Hashing Strategy

Use Python's dictionary (hash table) to store and retrieve word frequencies efficiently.

Data Structure

- **Key:** word
- **Value:** frequency

Suggested Steps

1. Preprocess text (lowercase, remove punctuation, split into words).
2. For each word:
    - If in dictionary → increment its count
    - Else → add with frequency 1
3. Sort results and print the 20 most frequent words

Example Output

Top 20 Word Frequencies

Word	Frequency
the	6523
and	4351
to	2987
…	…

Code Requirements

Your Python file must include a **docstring** with:

- Your name
- Student number
- Brief description of how the program works

Bonus Task (Optional, +5%) — Bigram Frequency Counter

Extend your program to also compute **bigram frequencies**.

A **bigram** is a pair of two consecutive words, e.g.:

- “in the”
- “to go”
- “we went”

Bonus Requirements

1. After counting individual words, generate all bigrams from the text.
2. Use a hash table where:
    - **Key:** a tuple or string representing the bigram
    - **Value:** frequency
3. Print the **10 most frequent bigrams**.

Example Bonus Output

Top 10 Bigrams

Bigram	Frequency
“of the”	354
“in the”	276
“to the”	245
…	…

Deliverables

Upload to Blackboard:

- Python file
- reflection.txt