Spam Email Filtering

Abstract-

This project implements a simple spam email filteingr program in C. The program identifies spam emails by scanning their content for predefined spam keywords stored in a hash table with linked list chaining. It processes the input email text dynamically and flags emails containing spam words. This approach allows efficient keyword lookup and demonstrates the use of fundamental data structures.

Problem Description-

Filtering spam emails is essential to prevent unwanted messages from inboxes. Manual detection is impractical, so automated techniques using keyword matching help identify spam quickly. This project aims to develop a basic spam detection system that reads email content and flags messages containing spam-related words.

Implementation Details-

- The program uses a hash table with chaining (linked lists) to store spam keywords for quick search.
- A simple hash function computes the index based on the sum of ASCII values of characters.
- The input email is read as a string, parsed word-by-word by extracting alphabetic sequences.
- Words are converted to lowercase for case-insensitive comparison.
- Each word is checked against the hash table; if a spam word is found, the email is classified as spam.
- Dynamic memory allocation (malloc) is used to handle input size and word extraction.
- Finally, allocated memory is freed properly to avoid leaks.

Data Structures Used-

- **Hash Table:** For storing spam words and fast lookups.
- **Linked Lists:** To handle collisions in the hash table through chaining.
- **Dynamic Arrays:** Used for processing and building words from the email input.

Challenges Faced-

- Implementing a reliable hash function that distributes spam keywords evenly.
- Ensuring proper cleanup of memory allocated for the hash table and linked lists.
- Designing the program to be simple yet efficient for the intended use case.

Conclusion and Learnings-

This project reinforced understanding of hash tables and linked lists in C, especially managing memory and pointers. It also highlighted the importance of efficient data structures in real-world problems like spam detection. Although basic, the program lays groundwork for more advanced filters involving machine learning or pattern matching. Future improvements could include reading emails from files, expanding the spam word list, and enhancing the hash function.