CSE2225 Data Structures

Project1 Report

150123014 Ali Doğan Pekdaş

1. Problem Definition

Suppose we have two positive decimal numbers that have infinitely many digits (a very long number). How can we multiply these two numbers? Write C code to solve this problem.

2. Implementation Details

First, I created the input file and read it. I included <time.h> function to evaluate the execution time. To do this, I extract the start time from the finish time. I implemented the product formula, I made it to write the multiplicand, multiplier, result and execution time to the output file.

After that I started the linked list structure. First I created a struct node containing digit and pointer itself, I took size of head pointer as malloc.

Functions for linked list:

- append: Adds new digits to structure.
- reverse: Takes digits with reverse order because digits are taken from headline and we make it to write in correct order with reverse function.
- addLists: Combines two numbers digit by digit and builds a new linked list.
- multiplyLists: Performs multiplication of two linked lists representing numbers, where each node contains a single digit. This function multiplies the two numbers digit by digit.
- readingFile: Reads two numbers (potentially with decimal points) from a file and populates two linked lists (multiplicand and multiplier) that represent these numbers. It also counts the number of digits after the decimal point for each number.
- printList: Displays the contents of a linked list, which represents a number, and formats the output properly.

In the main function, the program begins by recording the execution time, declaring linked list pointers for the multiplicand and multiplier, and initializing variables to track decimal places. It reads the numbers from a specified input file into linked lists using the readingFile function, then reverses the lists for processing. The program calls the multiplyLists function to compute the product of the two numbers, calculates the total decimal places, and reverses the result to restore the correct digit order. After stopping the timer and calculating execution time, it opens an output file to write and display the multiplicand, multiplier, result, and execution time using the printList function. Finally, the output file is closed, and the program exits successfully.