Project Details

In this algorithm, I initially created a struct to read numbers from a file. I intended to store the numbers and any decimal points using the ch value within the struct. Then, I added the values from the first line of the file to the linked list as multiplicand through current1. I did the same for the second line using multiplier and current2, and I placed the number storage and result calculation in a loop to perform multiple operations.

After retrieving the numbers, I sent them to the reverseList function to reverse them and then passed them to the multiply function for processing. The reason for reversing is related to starting the multiplication from the end of the number.

In the multiply function, I multiplied the obtained numbers digit by digit using nested while loops while ignoring the decimal points. When I multiplied each digit of the second number, I sent the resulting intermediate value to the addZeroList function to append the necessary number of zeros. I updated the finalResult by adding the new intermediate results to it. I also determined the position of the decimal point by shifting it in multiplicand and multiplier until I found it, and I inserted the decimal point into the finalResult linked list at the identified position to reach the most updated version of finalResult.

Function Explanations

- **number *reverseList(number *head)**: I used this to reverse the values in the linked list.
- **void printListForCheck(number *head)**: I used this for convenience in checking the correctness of the code's progress.
- **void printList(number *head, FILE *outputFile)**: I used this to write the values that need to be printed to the output file.
- **void multiply(number *multiplicand, number *multiplier, FILE *outputFile)**: I performed the multiplication operation in this function. Initially, I created a while loop for the multiplication process. I multiplied each digit of multiplicand with the digits of multiplier one by one, paying attention to carries. I sent the resulting intermediate value to reverseList to reverse it because it needed to be reversed to append zeros at the end. Then, I called addZeroList and appended the necessary zeroList at the end. After that, I obtained a finalResult by adding these intermediate

results to it in a loop. Finally, I reversed it again to insert the decimal point into the result. I searched for the positions of multiplicand and multiplier and moved through finalResult. I opened the linked list at the found position to insert the decimal point and then reconnected the linked list.

- **number *addLists(number *result1, number *result2)**: I used this to add each of the intermediate results to finalResult.
- **void addZeroList(number **result, number *zeroList)**: I used this to append the necessary number of zeros to the intermediate results. I added zeros to the previously created zeroList each time the digit of multiplier changed. This zeroList progressed as 0 -> 00 -> 000… and I added them sequentially to all intermediate results except the first.

Sultan Kocagöz

150123081