

**MARMARA UNIVERSITY**

**FACULTY OF ENGINEERING**

**CSE2025 Data Structures**

**PROJECT #1**

Due Date: 05.11.2021

**NAME SURNAME:** SERKAN KORKUT

**STUDENT ID:** 150119036

**Problem Definition**

Let's say we want to multiply infinitely large positive natural numbers, but we cannot multiply these numbers using standard data types in C. That’s why we will use linked lists to solve this problem.

**Functions**

* **void getMultiplicandMultiplier (numberNodePtr \*mltplcndPtr, numberNodePtr \*mltplrPtr):**

This function creates file pointer to read from input.txt file. In this file, there is a multiplicand on the first line and a multiplier on the second line. With fscanf function, these two numbers are read and transferred to multiplicand and multiplier strings. Firstly, each character of the multiplicand string is added to the multiplicand list, starting from the end. The same procedure is applied for the multiplier string. At the end of this function, two pointers are obtained, one for the multiplicand list and one for the multiplier list.

* **int numberOfDigits (numberNodePtr numberPtr):**

This function uses the node pointer inside the parameter and finds out how many elements are in that list. After that, the function returns this number.

* **int char2Int (char ch):**

This function has a character as its parameter. It gets a digit by subtracting the '0' character from this character. After that, it returns this number.

* **void node2String (char\* string, numberNodePtr numberPtr):**

This function takes character pointer and node pointer as parameters. With the for loop, it transfers the characters in the list one by one to the string. In this loop, the strrev function is used because the digits are placed in reverse order. As a result of this function, the desired string is obtained.

* **void writeOutput (numberNodePtr mltplcndPtr, numberNodePtr mltplrPtr, numberNodePtr resultPtr)**

This function creates a file pointer to write the multiplicand, multiplier and result to output.txt file. Each number is obtained with node2String function. Then, these numbers are printed to output.txt file with fprintf function.

* **numberNodePtr getResult (numberNodePtr mltplcndPtr, numberNodePtr mltplrPtr):**

This function uses two node pointers as parameters, one for the multiplicand and the other for the multiplier. The standard multiplication method is used for this operation. To implement this, two nested while loops are used.

****

2

4

6

8

****

4

8

2

3

****

6

9

4

****

****

7

2

0

2

3

****

****

1

8

1

9

5

****

8

7

6

5

9

1

1

Firstly, the first digit of the multiplier and all the digits of the multiplicand are multiplied one by one. When both two digits are multiplied, the result is transferred to product variable. The division of the product variable by 10 is assigned to remaining variable. Product % 10 is used in the new node created. Remainig variable is stored so it can be used for the next node. At the end of the outer loop, multiplicandPtr and currentPtr are returned to beginning of their lists, but at the end of the next loops, currentPtr moves one step forward in the list. If the number of multiplier digits is more than one, multiplication continues with the next multiplier digit. If the currentPtr exists, it updates that node. Otherwise, a new node is created. When the function is finished, the result list is complete and a pointer to the beginning of this list is returned.