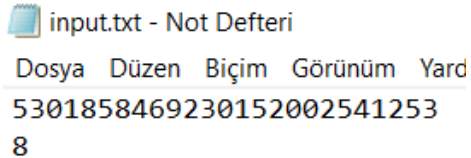


**CSE2025 Data Structures**  
**Project #1**  
***Merve Rana Kızıl – 150119825***

An input file which contains multiplicand and multiplier is read character by character so the program also can calculate the multiplication of infinitely large numbers.

Screenshot of a sample input file is given below.



input.txt - Not Defteri  
Dosya Düzen Biçim Görünüm Yard  
5301858469230152002541253  
8

malloc() function is used to allocate space from the memory to store multiplicand and multiplier in linked lists as required. Every digit of the numbers is kept in a data field of the linked lists.

Node\* multiply(Node\* multiplicand, Node\* multiplier) function is written which takes the heads of multiplicand and multiplier linked lists as inputs and returns the head of the result linked list.

Node\* result is defined, and its data fields are filled with 0's to ease the multiplication operation. To allocate the maximum length of the result linked list, the code snippet given below is used since the result of the multiplication of two numbers cannot be larger than total length of the two numbers + 1.


```
Node* result = initializeLinkedList(getLinkedListLength(multiplicand) + getLinkedListLength(multiplier) + 1);
```

Node\* intermediate\_result is defined to keep the temporary result during the multiplication operation. Multiplication of each two digits is kept in the variable currentResult. currentResult is divided by 10 so carry variable can be calculated. Modulus 10 of currentResult is used to fill the current digit of the intermediate\_result. This result is added to the current digit of the intermediate\_result which is initialized as 0. If the current digit of the intermediate\_result is greater than 10, a similar operation is repeated. Multiplicand and multiplier are traversed to implement the operation, nested while loops are used for this. multiplicand is traversed in the inner loop and the multiplier traversed in the outer loop. step variable is used to move the intermediate\_result after multiplying a digit of the multiplier and the digits of the multiplicand is finished. step variable is incremented by one in the outer loop.

At the end of the operation and printing the multiplicand, multiplier, and the result into the file, memory is freed by using the free() function. void freeLinkedList(Node\* head) is written to free the memory.

An output file is created after the run of the program. Below, a screenshot of the output file is given.

---

 output.txt - Not Defteri  
Dosya Düzen Biçim Görünüm Yardım  
multiplicand: 5301858469230152002541253  
multiplier: 8  
result: 42414867753841216020330024