- **Identify at least 1 edge case(s)** for the problem and expected results

  Input strings contain non-numeric characters as well.

  Expected Output: Program should inform user that an invalid number has been added and print empty multiplication result

- **Describe 3 test case**(s) and expected result(s) for the program
  1. Sample input.

     12345678945612345698754875464
     65498765432165498765456459875654
     Expected Output:
     808626729359487311150260374941183571521561714156923695553456

  2. Empty Input

     1234 and “”
     Expected Output: “”

  3. Negative numbers are entered
     -123 and 101
     Expected Output: “Invalid number entered.”

- **Describe** your algorithm in words
    1. Take both numbers as String input from User. Check if each number is a valid numeric string; if either is an empty string negative or contains non-numeric characters, inform the user and do not progress with multiplication.
    2. If both numbers are valid, proceed with the computeMultiplication function.
        a. Convert each numeric string into Node lists in reverse order.
        b. Initialize an answer list which will contain string lengths number of nodes each initialized as 0. (Based on numbers entered, multiplication can have maximum digits as sum of length of both numbers)
        c. For each digit ‘i’ in 2nd number:
            i. Multiply it with the entire first number
            ii. Start storing the result from ith node in the result node.
            iii. While multiplying each digit in the first number with the ith digit in the 2nd number, add carry-over and existing value at current node.
            iv. After multiplying with entire first number, store remaining carry in the result
        d. Repeat step c for all digits in 2nd number
    3. After computing multiplication and storing it in the result node, print the result. Iterate over the result node list and add each digit to the beginning of StringBuilder. Print the result stored in String.

- **Explain** your algorithm (provide a proof sketch of why it works)

    Aim of this program is to:

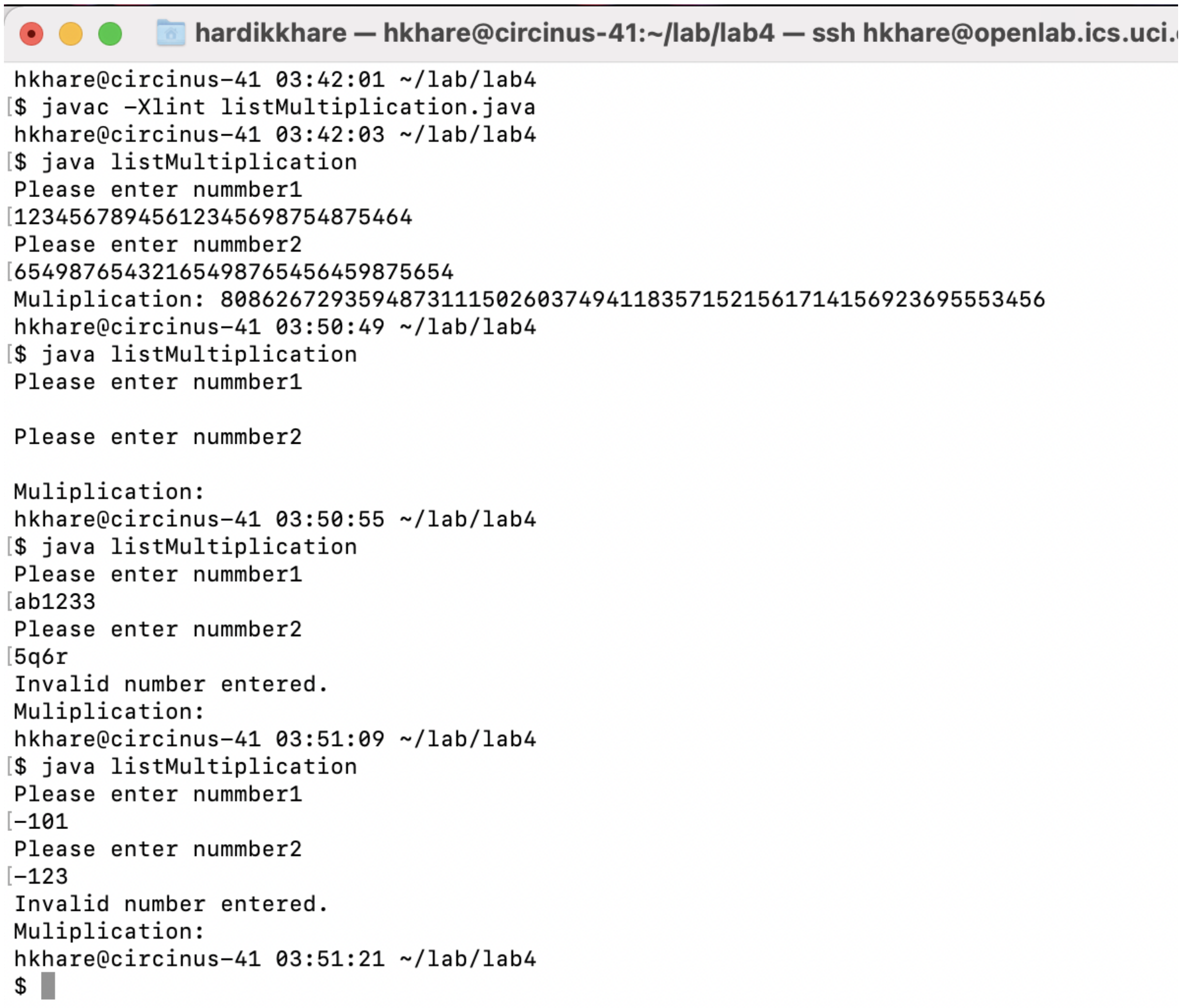
    - **Take numeric input from the user as strings:** To achieve this, we are storing input numeric strings in reverse order in a list of nodes.
    - **Perform multiplication and store the result in a list of nodes:** To achieve this, we are multiplying each digit in 2nd number with the entire 1st number. Each of this entire result is added to existing nodes at corresponding positions (keeping track of multiplication place offset at each step) in the result node list. Carry is also properly stored.
    - **Display multiplication output:** To achieve this, node list is iterated and digits are printed in reverse order.

- **Analyze** the **time** and space **complexity** of the solution. Explain why the time and space complexity fit your algorithm.

  Overall time complexity is O(mn). This is because we are multiplying every digit in the 2nd number with every digit in the 1st number.

  Space complexity will be O(m+n). For multiplying 2 numbers with length m and n, the max number of digits in the result can be m+n only.

- **Test**, and **demonstrate** correct function of the solution (add screenshots of your test cases)



hardikkhare — hkhare@circinus-41:~/lab/lab4 — ssh hkhare@openlab.ics.uci.

```
 hkhare@circinus-41 03:42:01 ~/lab/lab4
[$ javac -Xlint listMultiplication.java
 hkhare@circinus-41 03:42:03 ~/lab/lab4
[$ java listMultiplication
 Please enter nummber1
[12345678945612345698754875464
 Please enter nummber2
[654987654321654987654564598756 54
 Muliplication: 808626729359487311150260374941183571521561714156923695553456
 hkhare@circinus-41 03:50:49 ~/lab/lab4
[$ java listMultiplication
 Please enter nummber1

 Please enter nummber2

 Muliplication:
 hkhare@circinus-41 03:50:55 ~/lab/lab4
[$ java listMultiplication
 Please enter nummber1
[ab1233
 Please enter nummber2
[5q6r
 Invalid number entered.
 Muliplication:
 hkhare@circinus-41 03:51:09 ~/lab/lab4
[$ java listMultiplication
 Please enter nummber1
[-101
 Please enter nummber2
[-123
 Invalid number entered.
 Muliplication:
 hkhare@circinus-41 03:51:21 ~/lab/lab4
 $ █
```