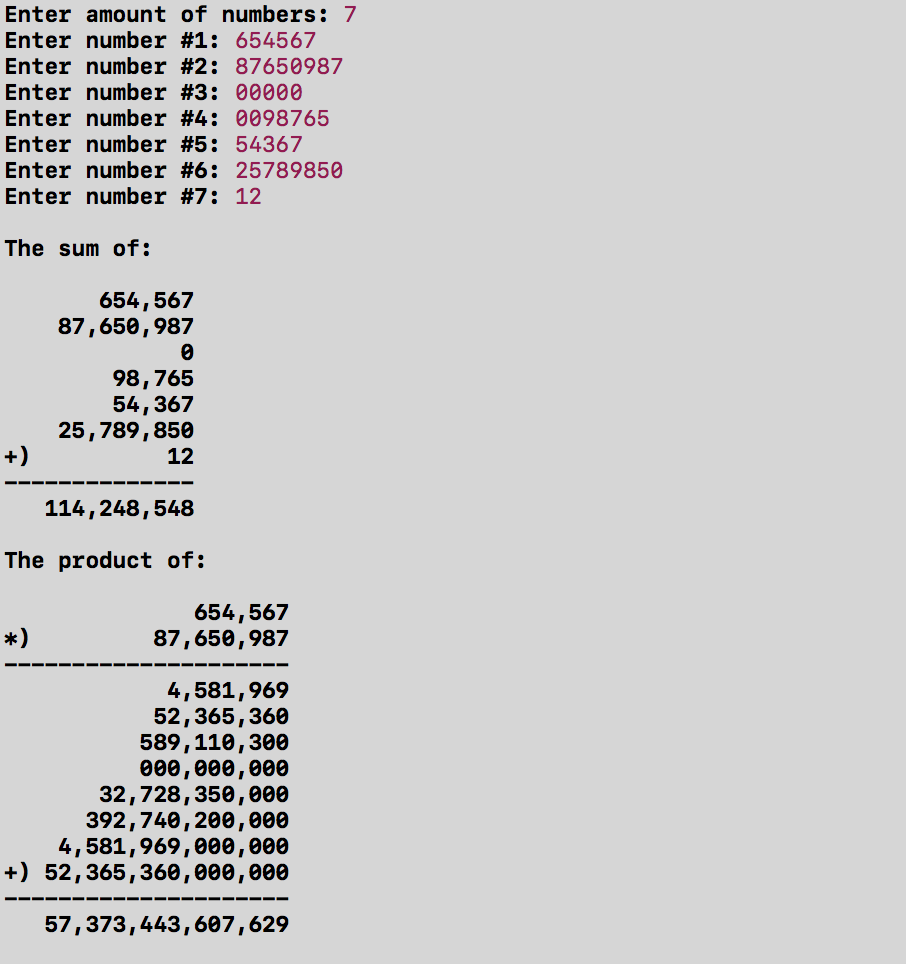
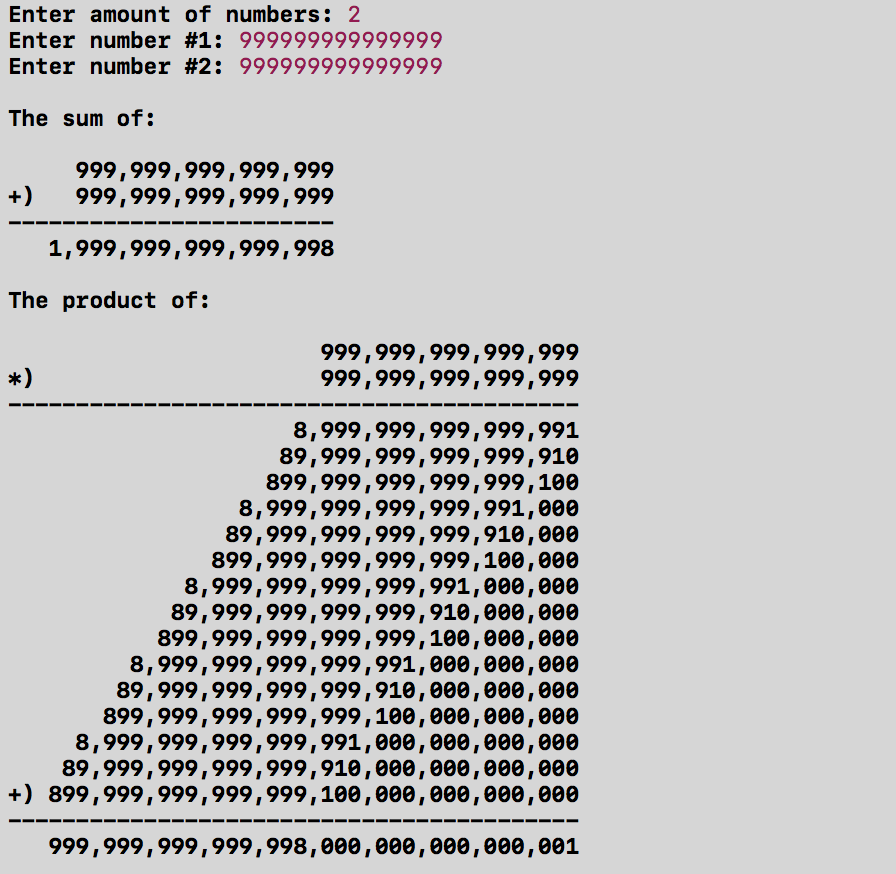
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Step | Objective | Test Data | Expected Outcome | Actual Outcome | Result |
| 1 | ValidateAmount checks bounds | Input: ‘m’, 1, and 11 | Invalid input | Invalid input | Pass |
| 2 | ValidateNumber check bounds | Input: 16 digit long number | Invalid input | Invalid input | Pass |
| 3 | SetNumber  format | 00000  00987  3456 | 0  987  3,456 | 0  987  3,456 | Pass |
| 4 | AddNumbers | 100  233  9999 | 10,332 | 10,332 | Pass |
| 4.5 | Number.AddNumber | 332  1 | 333 | 333 | Pass |
| 5 | MultiplyFirstTwoNumbers | 10  36 | 360 | 360 | Pass |
| 5.1 | MultiplyFirstTwoNumbers  Edge case | 15 9’s  15 9’s | 999,999,999,999,998  ,000,000,000,000,001 | 999,999,999,999,998  ,000,000,000,000,001 | Pass |
| 6 | FormatNumber | 0  5678  678 | 0  5,678  678 | 0  5,678  678 | Pass |
| 7 | PrintNumbers  alignment | 5  5432  65 | Numbers left aligned with respect to given length | Numbers left aligned with respect to given length | Pass |
| 8 | ValidatePosition  occupied space | “a1”..”a1” | Invalid position | Invalid position | Pass |

**Number Format Checking**

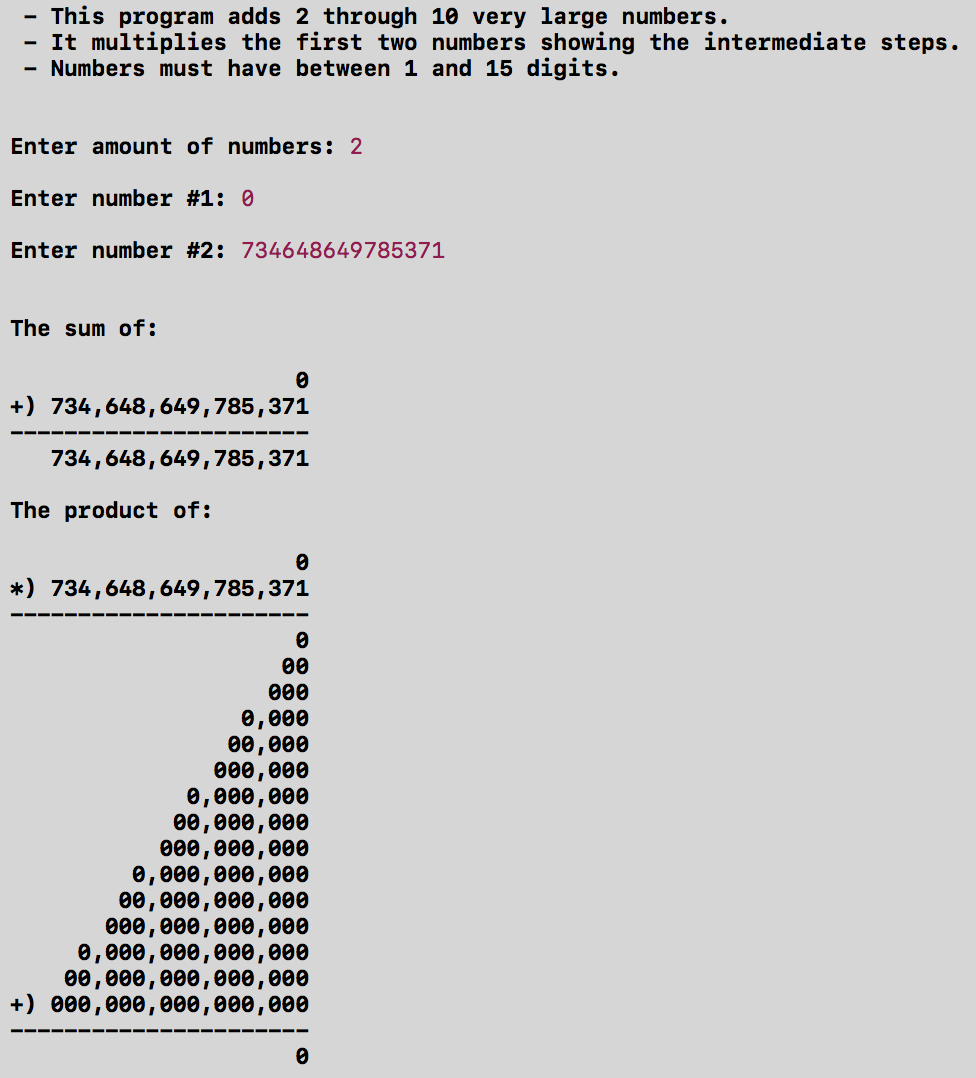
****

* Input “00000” is formatted as “0”.
* Input “0098765” is formatted as “98,765”
* Addition is correct and left-aligned with respect to sum length.
* Multiplication is correct and left-aligned with respect to product length.
* The intermediate steps of multiplication is correct and is proper aligned digit-by-digit
* Length of line is correct.
* Symbols are in the correct positions.

**Checking Edge Case**



* Remainder is properly carried over.
* Addition is correct and left-aligned with respect to sum length.
* Multiplication is correct and left-aligned with respect to product length.
* The intermediate steps of multiplication is correct and is proper aligned digit-by-digit.
* Product is properly displayed and fits into number\_array.

 **Final Format**

* Program information is displayed with correct numbers.
* Good amount of spacing between outputs.
* Addition is correct and left-aligned with respect to sum length.
* Multiplication is correct and left-aligned with respect to product length.
* The intermediate steps of multiplication is correct and is proper aligned digit-by-digit.
* Product is displayed as 0.