**Multiply Strings**

Question

Given two non-negative integers num1 and num2 represented as strings, return the product of num1 and num2, also represented as a string.

**Note:** You must not use any built-in BigInteger library or convert the inputs to integer directly.

**Example 1:**

**Input:** num1 = "2", num2 = "3"

**Output:** "6"

**Example 2:**

**Input:** num1 = "123", num2 = "456"

**Output:** "56088"

**Constraints:**

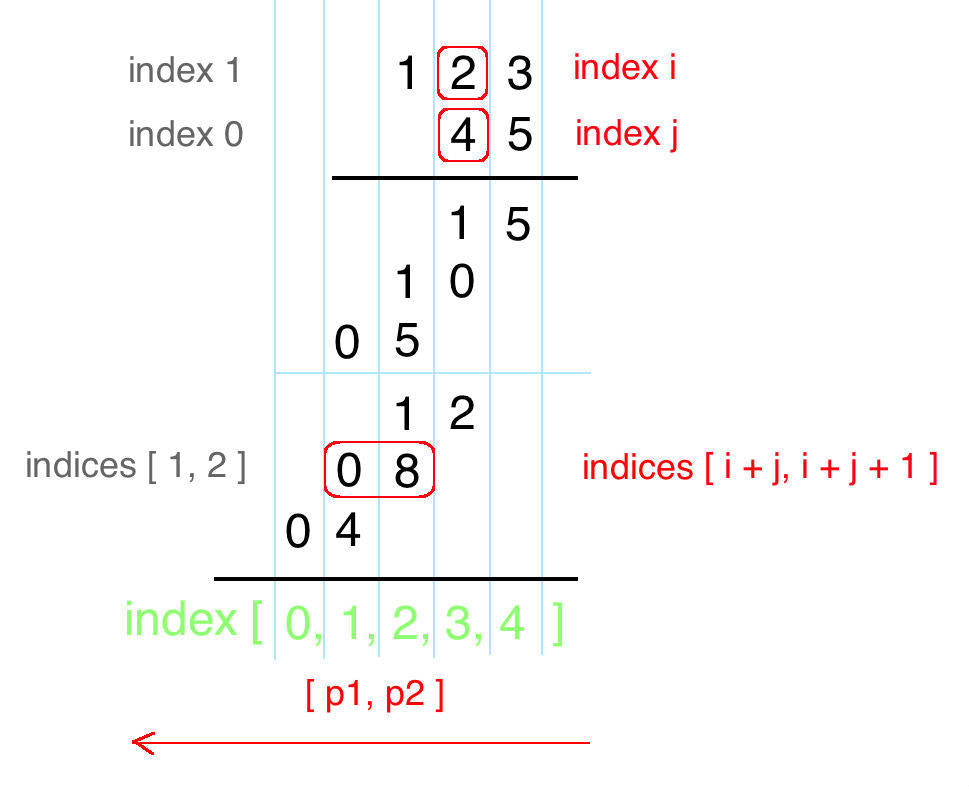
* 1 <= num1.length, num2.length <= 200
* num1 and num2 consist of digits only.
* Both num1 and num2 do not contain any leading zero, except the number 0 itself.

## Solution

Remember how we do multiplication?

Start from right to left, perform multiplication on every pair of digits, and add them together. Let's draw the process! From the following draft, we can immediately conclude:

num1[i] \* num2[j]` will be placed at indices `[i + j`, `i + j + 1]`



**Coding Solution**

Java

|  |
| --- |
| public String multiply(String num1, String num2) {  int m = num1.length(), n = num2.length();  int[] pos = new int[m + n];    for(int i = m - 1; i >= 0; i--) {  for(int j = n - 1; j >= 0; j--) {  int mul = (num1.charAt(i) - '0') \* (num2.charAt(j) - '0');  int p1 = i + j, p2 = i + j + 1;  int sum = mul + pos[p2];  pos[p1] += sum / 10;  pos[p2] = (sum) % 10;  }  }    StringBuilder sb = new StringBuilder();  for(int p : pos) if(!(sb.length() == 0 && p == 0)) sb.append(p);  return sb.length() == 0 ? "0" : sb.toString();  } |