Plus One

You are given a **large integer** represented as an integer array digits, where each digits[i] is the ith digit of the integer. The digits are ordered from most significant to least significant in left-to-right order. The large integer does not contain any leading 0's.

Increment the large integer by one and return *the resulting array of digits*.

**Example 1:**

**Input:** digits = [1,2,3]

**Output:** [1,2,4]

**Explanation:** The array represents the integer 123.

Incrementing by one gives 123 + 1 = 124.

Thus, the result should be [1,2,4].

**Example 2:**

**Input:** digits = [4,3,2,1]

**Output:** [4,3,2,2]

**Explanation:** The array represents the integer 4321.

Incrementing by one gives 4321 + 1 = 4322.

Thus, the result should be [4,3,2,2].

**Example 3:**

**Input:** digits = [9]

**Output:** [1,0]

**Explanation:** The array represents the integer 9.

Incrementing by one gives 9 + 1 = 10.

Thus, the result should be [1,0].

**Constraints:**

* 1 <= digits.length <= 100
* 0 <= digits[i] <= 9
* digits does not contain any leading 0's.

Solution

public class Solution {

    public int[] PlusOne(int[] digits) {

        // Reading the digits into a string num

        string num ="";

        for(int i= 0; i<digits.Length; i++){

            num += digits[i].ToString();

        }

        // Incrementing By 1

        BigInteger digit = 0;

        digit = BigInteger.Parse(num);

        digit = digit + 1;

        // Converting back the digit to string

        num = digit.ToString();

        // Array to copy the new digits

        int[] output = new int[num.Length];

        for(int i = 0; i < num.Length; i++){

            int ans = num[i] - '0';

            output[i] = ans;

        }

        return output;

    }

}