

[1031 Add to Array-Form of Integer \(link\)](#)

Description

The **array-form** of an integer `num` is an array representing its digits in left to right order.

- For example, for `num = 1321`, the array form is `[1,3,2,1]`.

Given `num`, the **array-form** of an integer, and an integer `k`, return *the array-form of the integer* `num + k`.

Example 1:

Input: `num = [1,2,0,0]`, `k = 34`
Output: `[1,2,3,4]`
Explanation: `1200 + 34 = 1234`

Example 2:

Input: `num = [2,7,4]`, `k = 181`
Output: `[4,5,5]`
Explanation: `274 + 181 = 455`

Example 3:

Input: `num = [2,1,5]`, `k = 806`
Output: `[1,0,2,1]`
Explanation: `215 + 806 = 1021`

Constraints:

- $1 \leq \text{num.length} \leq 10^4$
- $0 \leq \text{num}[i] \leq 9$
- `num` does not contain any leading zeros except for the zero itself.
- $1 \leq k \leq 10^4$

(scroll down for solution)

Solution

Language: *cpp*

Status: Accepted

```
#include <vector>
using namespace std;

class Solution {
public:
    vector<int> addToArrayForm(vector<int>& num, int k) {
        vector<int> result;
        int carry = 0;
        int i = num.size() - 1;

        while (i >= 0 || k > 0 || carry > 0) {
            int sum = carry;
            if (i >= 0) {
                sum += num[i];
            }
            if (k > 0) {
                sum += k % 10;
                k /= 10;
            }

            result.push_back(sum % 10);
            carry = sum / 10;

            i--;
        }

        reverse(result.begin(), result.end());
        return result;
    }
};
```