

Add Strings

Given two non-negative numbers `num1` and `num2` represented as string, return the sum of `num1` and `num2`.

Note:

1. The length of both `num1` and `num2` is
2. Both `num1` and `num2` contains only digits `0-9`.
3. Both `num1` and `num2` does not contain any leading zero.
4. You **must not use any built-in BigInteger library** or **convert the inputs to integer** directly.

Solution 1

```
class Solution {
public:
    string addStrings(string num1, string num2) {
        int n1 = num1.size(), i = n1 - 1;
        int n2 = num2.size(), j = n2 - 1;
        int carry = 0;
        string res = "";
        while(i >= 0 || j >= 0 || carry){
            long sum = 0;
            if(i >= 0){sum += (num1[i] - '0'); i--;}
            if(j >= 0){sum += (num2[j] - '0'); j--;}
            sum += carry;
            carry = sum / 10;
            sum = sum % 10;
            res = to_string(sum) + res;
        }
        return res;
    }
};
```

written by [jasonshieh](#) original link [here](#)

Solution 2

```
public String addStrings(String num1, String num2) {
    int len1=num1.length()-1;
    int len2=num2.length()-1;

    StringBuilder sb=new StringBuilder();
    int sum=0,carry=0;
    while(len1>=0 || len2>=0) {
        int first=len1>=0?num1.charAt(len1)-'0':0;
        int second=len2>=0?num2.charAt(len2)-'0':0;
        sum=carry+first+second;
        if(sum<=9){
            sb.insert(0,sum);
            sum=0;
            carry=0;
        } else {
            sb.insert(0,sum%10);
            sum=0;
            carry=1;
        }
        len1--;
        len2--;
    }
    if(carry==1)sb.insert(0,"1");
    return sb.toString();
}
```

written by [guruvignesh](#) original link [here](#)

Solution 3

```
string addStrings(string num1, string num2) {
    int len1 = num1.size(), len2 = num2.size(), add = 0;
    if (len1 < len2) return addStrings(num2, num1);
    for (int i = 1; i <= len1 && (add || i <= len2); i++, add /= 10) {
        add += num1[len1 - i] - '0' + (i <= len2 ? num2[len2 - i] - '0' : 0);
        num1[len1 - i] = add % 10 + '0';
    }
    return (add ? to_string(add) : "") + num1;
}
```

written by [zyoppyoo8](#) original link [here](#)

From [LeetCoder](#).