

字符串相加

给定两个字符串形式的非负整数 num_1 和 num_2 ，计算它们的和。

提示：

1. num_1 和 num_2 的长度都小于 5100
2. num_1 和 num_2 都只包含数字 0-9
3. num_1 和 num_2 都不包含任何前导零
4. 你不能使用任何内建 **BigInteger** 库，也不能直接将输入的字符串转换为整数形式



具体实现也不复杂，我们定义两个指针 i 和 j 分别指向 num_1 和 num_2 的末尾，即最低位，同时定义一个变量 add 维护当前是否有进位，然后从末尾到开头逐位相加即可。你可能会想两个数字位数不同怎么处理，这里我们统一在指针当前下标处于负数的时候返回 0，等价于**对位数较短的数字进行了补零操作**，这样就可以除去两个数字位数不同情况的处理，具体可以看下面的代码。

```
class Solution {
public:
    string addStrings(string num1, string num2) {
        int s1=num1.size()-1;
        int s2=num2.size()-1;//记录下标 逆序遍历 逐位相加
        int add=0;//进位
        string res="";
        while(s1>=0 || s2>=0 || add!=0)
        {
            int x=s1>=0?num1[s1]-'0':0;
            int y=s2>=0?num2[s2]-'0':0;
            int result=x+y+add;
            res.push_back(result%10+'0');
            add=result/10;
            s1--;
            s2--;
        }
        reverse(res.begin(),res.end());//逆序翻转
        return res;
    }
};
```