

给定一个**非负整数**组成的**非空**数组，在该数的基础上加一，返回一个新的数组。

最高位数字存放在数组的首位， 数组中每个元素只存储一个数字。

你可以假设除了整数 0 之外，这个整数不会以零开头。

示例 1:

输入: [1,2,3]

输出: [1,2,4]

解释: 输入数组表示数字 123。

示例 2:

输入: [4,3,2,1]

输出: [4,3,2,2]

解释: 输入数组表示数字 4321。

思路：暴力解，反正是+1操作，统计个位连续的9就行，9之前全为0，9之后第一位+1，后面不变。

```
class Solution {
public:
    vector<int> plusOne(vector<int>& digits) {
        vector<int> result;
        int num=0;
        int n=digits.size();
        int i;
        for(i=n-1;i>=0&&digits[i]==9;--i){
        }
        if(i<0){
            result.push_back(1);
            for(int j=0;j<n;++j){
                result.push_back(0);
            }
        }else{
            for(int j=0;j<i;j++){
                result.push_back(digits[j]);
            }
            result.push_back(digits[i]+1);
            for(int j=i+1;j<n;j++){
                result.push_back(0);
            }
        }
        return result;
    }
};
```