

加一

给定一个由 **整数** 组成的 **非空** 数组所表示的非负整数，在该数的基础上加一。

最高位数字存放在数组的首位， 数组中每个元素只存储**单个**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: digits = [1,2,3]

输出: [1,2,4]

解释: 输入数组表示数字 123。

示例 2:

输入: digits = [4,3,2,1]

输出: [4,3,2,2]

解释: 输入数组表示数字 4321。

示例 3:

输入: digits = [0]

输出: [1]

```
/**
 * Note: The returned array must be malloced, assume caller calls free().
 */
/*测试用例数据过大，超出存储范围
int* plusOne(int* digits, int digitsSize, int* returnSize){
    long long data=0;
    int flag=1;
    for(int i=0;i<digitsSize;i++)
    {
        data=data*10+digits[i];
    }
    int* arr=(int*)malloc(sizeof(int)*(digitsSize+1));
    memset(arr,0,sizeof(int)*(digitsSize+1));
    long long res=data+1;
    for(int j=digitsSize;j>=0;j--)
    {
        arr[j]=res%10;
```

```

        res/=10;
    }
    if(arr[0]==0)
    {
        *returnSize=digitsSize;
        return arr+1;
    }
    else
    {
        *returnSize=digitsSize+1;
        return arr;
    }
}*/

/**
 * Note: The returned array must be malloced, assume caller calls free().
 */
//从后向前(从个位)开始+1,逢九进一,不为九时结束,全为九时数组长度加一
int* plusOne(int* digits, int digitsSize, int* returnSize)
{
    for(int i=digitsSize-1;i>=0;i--)
    {
        if(digits[i]==9)
        {
            digits[i]=0;
        }
        else{
            digits[i]++;
            *returnSize=digitsSize;
            return digits;
        }
    }
    //此时说明全为0 需要进位 数组加长 1
    int* arr=(int*)malloc(sizeof(int)*(digitsSize+1));
    memset(arr,0,sizeof(int)*(digitsSize+1));
    arr[0]=1;
    *returnSize=digitsSize+1;
    return arr;
}

```