**1、：反转字符串**

**编写一个函数，其作用是将输入的字符串反转过来。输入字符串以字符数组 “char[]” 的形式给出。**

**不要给另外的数组分配额外的空间，你必须\*\*[原地]修改输入数组\*\*、使用 O(1) 的额外空间解决这一问题。**

**所有字符都是 [ASCII] 码表中的可打印字符。**

**示例 1：**

**输入：["h","e","l","l","o"]**

**输出：["o","l","l","e","h"]**

**2二进制中1的个数**

**请编写一个程序，输入一个十进制整数，输出该数二进制表示中 1 的个数。例如，把 9 表示成二进制是 1001，有 2 位是 1。因此，如果输入 9，则该函数输出 2。（测试时十进制整数的范围不超过1023且不低于-1023）**

**示例 1：**

**输入：11**

**输出：3**

**解释：二进制串 0000001011 中，共有三位为 '1'。**

**示例 2：**

**输入：128**

**输出：1**

**解释：二进制串 0010000000 中，共有一位为 '1'。**

**3、：给定一个 没有重复 数字的序列，返回其所有可能的全排列。**

**示例:**

**输入: [1,2,3]**

**输出:**

**[**

**[1,2,3],**

**[1,3,2],**

**[2,1,3],**

**[2,3,1],**

**[3,1,2],**

**[3,2,1]**

**]**

**4、：字符串相加**

**给定两个字符串形式的非负整数 num1 和num2 ，计算它们的和。**

**提示：num1 和num2 的长度都小于 5100**

**num1 和num2 都只包含数字 0-9**

**num1 和num2 都不包含任何前导零**

**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**

**5、：删除链表的倒数第N个节点**

**给定一个链表，删除链表的倒数第 n 个节点，并且返回链表的头结点。**

**示例 1：**

**给定一个链表: 1->2->3->4->5, 和 n = 2.**

**当删除了倒数第二个节点后，链表变为 1->2->3->5.**

**说明：**

**给定的 n 保证是有效的。**

**进阶：**

**你能尝试使用一趟扫描实现吗？**

**6、：从前序与中序遍历序列构造二叉树**

**根据一棵树的前序遍历与中序遍历构造二叉树。**

**注意:**

**你可以假设树中没有重复的元素。**

**例如，给出：**

**前序遍历 preorder = [3,9,20,15,7]**

**中序遍历 inorder = [9,3,15,20,7]**

**返回如下的二叉树：**

**3**

**/ \**

**9 20**

**/ \**

**15 7**

**7.** **给定一个无重复元素的数组 candidates 和一个目标数 target ，找出 candidates 中所有可以使数字和为 target 的组合。**

**candidates 中的数字可以无限制重复被选取。**

**说明：**

**所有数字（包括 target）都是正整数。**

**解集不能包含重复的组合。**

**示例 1：**

**输入：candidates = [2,3,6,7], target = 7,**

**所求解集为：**

**[**

**[7],**

**[2,2,3]**

**]**

**示例 2：**

**输入：candidates = [2,3,5], target = 8,**

**所求解集为：**

**[**

**[2,2,2,2],**

**[2,3,3],**

**[3,5]**

**]**