第一题：

页面截图：

题目：编程实现功能：
功能是求出字符串 s 与字符串 t 的第二长公共单词(这里，假设两个字符串均由英文字母和空格字符组成)；若找到这样的公共单词，函数返回该单词，否则，函数返回NULL，如果有多个满足要求，则返回第一个单词。
例如：若 s=“This is C programming text”，t=“This is a text for C programming”则输出结果为 'is'

S: This is C programming text

T: This is a text for C programming 输出

显示结果：

Html 代码：

```
S: <input type="text" id="inputs" style="width: 300px;height: 30px;"
value="This is C programming text"></br>
T: <input type="text" id="inputt" style="width: 300px;height: 30px;"
value="This is a text for C programming">
<input type="button" value="输出" onclick="putOutWord()" style="width:
60px;height: 30px;">
<div class="show0">显示结果：</div>
```

Js 代码：

```
    <script>
        // 第一题
        // 思路：
        // 1. 找出公共单词即单词长度组成新的对象数组
        // 2. 对新对象数组排序去重得到新的数组
        // 3. 索引新数组的第二个元素，即第二长公共单词
        function putOutWord() {
            var inputs = document.getElementById('inputs').value;
            var inputt = document.getElementById('inputt').value;
            var arr_inputs = inputs.split(' ');
            var same = [];
            var p = 0;
            // console.log(arr_inputs);
            for (i = 0; i < arr_inputs.length;i++) { // 生成对象数组
                if (inputt.indexOf(arr_inputs[i]) !== -1) {
                    // alert(arr_inputs[i])
                   var item = {
                        word: arr_inputs[i],
                        len: arr_inputs[i].length
                    }
                    same.push(item)
                }
            }
            // console.log(same);
```

```
        // console.log(same);
            same.sort(function(item1,item2){ // 重写排序
                var one = item1.len;
                var two = item2.len;
                if (one < two) {
                    return -1;
                } else if (one > two) {
                    return 1;
                } else {
                    item2 = null;
                    return 0;
                }
            })
            var toBeSiggle = function (same) { // 去重
                for (k = 0; k < same.length; k++) {
                    if (same.length < 2) {
                        alert ('不存在第二长公共单词')
                    } else {
                        var siggle=[same[0]];
                        for(var l = 1; l < same.length; l++){
                          if( same[l].len !== siggle[siggle.length-1].len) {
                          siggle.push(same[l]);
                         }
                        }
                     }
                     return siggle
                    }
                }
                var result = toBeSiggle(same)
                document.getElementsByClassName('show0')[0].innerHTML = "显示结果：  " + result[1].word;
        }

        // 第二题
```

第二题：

页面截图：

题目：
某些整数能分解成若干个连续整数的和的形式，例如
15 = 1 + 2+3+4+5
15 = 4 + 5 + 6
15 = 7 + 8
某些整数不能分解为连续整数的和，例如：16
输入：一个整数N（N <= 10000）
输出：整数N对应的所有分解组合，按照每个分解中的最小整数从小到大输出，每个分解占一行，每个数字之间有一个空格（每行最后保留一个空格）；如果没有任何分解组合，则输出NONE.

9 输出

显示结果：
2 3 4
4 5

Html 代码：

```
<input type="text" id="input" style="width: 200px;height: 30px;"
value="15">
<input type="button" value="输出" onclick="putOut()" style="width:
60px;height: 30px;">
<div class="show">显示结果：</div>
```

Js 代码：

```
    // 第二题
    // 思路：
    // 1. 找出连续整数相加和等于输入值所有组合，组成数组
    // 2. 将数组显示在页面
    function putOut () {
        document.getElementsByClassName('show')[0].innerHTML = "显示结果："
        var input = document.getElementById('input').value;
        if (input <= 10000) {
            var a = 0;
            var n = 0;
            var arr_sum = [];
            for (j = 1;j <= input; j++) {
                n = j;
                arr_sum[a] = [];
                var sum = 0;
                var zancun = [];  // 定义一个变量暂存1--n的整数
               for (; n <= input; n++) {
                   var sum = sum + n; // 记录1--n相加的和
                   if (sum <= input) {
                       zancun.push(n);
                       if (sum == input) { // 和等于输入值的时，停止暂存n，将和恰好等于输入值的这组整数数组保存下来
                           arr_sum[a++] = zancun;  // arr_sum 表示满足条件的整数的数组的集合
                       }
                   }
               }
            }
        } else {
            alert ("请输入10000以内的数")
        }
        if (arr_sum.length <= 1) {
            document.getElementsByClassName('show')[0].innerHTML = document.getElementsByClassName('show')[0].innerHTML + ' NONE';
        } else {
            for (let k = 0; k < arr_sum.length-1; k++) {
               var show = document.querySelector(".show");
               var con = document.createElement("div");
               con.innerHTML = arr_sum[k].join(' ')
               show.appendChild(con);
            }
        }
    }
</script>
```