

代码逻辑训练题

(第 8 次)

限时：90 分钟

注：本次测试题为 2 题

- 文件编码必须为 UTF-8
 - Java 只允许使用 JDK 中的 API，不允许使用第三方 jar；Java 版本限定 1.6。
 - JavaScript 只允许使用原生，不要引入第三方 js 文件，且最终以 js 文件方式提交，不要将结果写在 html 中。
- 如违反上述条例均以签名错误判定。

题目一

问题描述

大多数 RPG 游戏的特点是，随着游戏的进展，你的英雄需要通过打怪获得经验来进行升级，以使你的英雄变得更强。因此你的游戏向导需要告诉你还需要多少经验你的英雄才能升到下一级。给定一个经验数组 `int[] expNeeded`。表示升到各级需要的经验值。举例说明，如果 `expNeeded` 为{150, 450, 900, 1800}，那么你需要 150 经验从 level 0 升级到 level 1，需要 300 经验(450 - 150)，从 level 1 升级到 level 2，以此类推。

初始的时候，你的英雄的经验值为 0，每完成一些任务，你的英雄会获得一些经验。你随时想知道离下一次升级还需要多少经验值。给定 `int[]` 参数 `expNeeded`，表示升到各级需要的经验值，给定一个 `int` 参数 `received` 表示当前已获得的经验，返回一个 `int` 值，表示离下次升级还需要多少经验。

约束

1. 参数 `expNeeded` 只会包含 1 到 50 个元素（包含 1 和 50），并且元素值只会以升序排列。
2. 参数 `expNeeded` 中元素的数值从 1 到 999999（包含 1 和 999999）
3. 参数 `received` 的值在 0 和 `expNeeded` 元素的最大值之间，即 $0 \leq \text{received} < \text{expNeeded}$ 中元素的最大值

示例

输入	返回
{150,450,900,1800}	17

133	说明：升级到 level 1 还需要 17 的经验
{150,450,900,1800} 312	138 说明：升级到 level 2 还需要 138 的经验
{150,450,900,1800} 612	288
{150,450,900,1800} 450	450

定义

Java

包名：	自己名字的缩写，如：package lhg;
类名：	LevelUp
方法：	toNextLevel
参数：	int[],int
返回值：	int
方法签名：	public int toNextLevel(int[] expNeeded, int received)

JavaScript

文件名：	LevelUp.js
函数名：	toNextLevel
参数：	数字数组,数字
返回值：	数字
方法签名：	function toNextLevel (expNeeded, received)

题目二

问题描述

你是一所学校的体育教练，你将在学校举办一场团体越野赛事，并记录赛事成绩。考虑到手工统计比赛名次非常繁琐，你决定编写一个程序来帮助统计。给定参赛队伍的数量和队伍中队员的名次，统计最终的队伍排名。

各个参赛队伍将用大写字母表示，比如第一支队伍用‘A’表示，第二支队伍用‘B’，以此类推。每支队伍前 5 名完赛的队员的成绩（比如第一名成绩为 1，第二名成绩为 2，以此类推）将加在一起作为队伍的总成绩。队伍的总成绩得分越低，则排名越靠前。如果队伍完赛的队员不足 5 人，则该队伍成绩忽略。如果两支队伍成绩相同，则比较两队第 6 名完赛的队员的成绩，第 6 名队员率先完赛的队伍排名靠前。如果两支队伍成绩相同，但其中只有一队有第 6 名队员的成绩，则该队排名靠前。如果两支队伍成绩相同，但都没有第 6 名队员的成绩，则队名按字母顺序靠前的队伍排名靠前（比如 A 队和 C 队成绩相同，则该情况下 A 队排名靠前）。

给定一个 `int` 参数 `numTeams`，表示参赛队伍的数量，给定一个 `String` 参数 `finishOrder`，表示所有参赛队员的完成名次，请返回最终的队伍排名，请注意，如果一支队伍的完赛队员不足 5 人，请忽略这支队伍，不用参加队伍排名，即该队不必包含在返回值中。

举例：如果 `finishOrder="AABBABBABBA"`，则队伍 A 队员的完赛成绩为：第 1，第 2，第 5，第 8 和第 11，则队伍 A 的总成绩为 $1+2+5+8+11=27$ 分，队伍 B 队员的完赛成绩为：第 3，第 4，第 6，第 7，第 9 和第 10，则队伍 B 的总成绩为 $3+4+6+7+9=29$ 分（注意队伍成绩只累加前 5 名队员的成绩），所以 A 队排名靠前，返回“AB”

约束

1. `numTeams` 的数值只会在 2 到 10 之间（包含 2 和 10）
2. `finishOrder` 只会包含 10 到 50 个字符（包含 10 和 50），并且每个字符只会是大写的字母。

限制

运行时间限制：执行方法的计算时间不能超过 2 秒。

示例

输入	返回
----	----

2 "AABBABBABBA"	"AB"
3 "CCCB BBBBAAAACC"	"BC" 说明：队伍 A 只有 3 名完赛者，不足 5 人，不计入 A 的成绩。队伍 B 的成绩为 30，队伍 C 的成绩为 31，队伍 B 排名靠前，返回"BC"。
4 "ABDCBADBDCCDBCDBCAAAC"	"BDCA"
10 "BDHCEAJBIDBCCHGCB DJEBAAHEGAGGADHGIE CJEHAEBDADJCDHG"	"BCDAHEG"
3 "BABCAABABAB"	"AB"

定义

Java

包名：	自己名字的缩写，如：package lhg;
类名：	CrossCountry
方法：	scoreMeet
参数：	int, String
返回值：	String
方法签名：	public String scoreMeet(int numTeams, String finishOrder)

JavaScript

文件名：	CrossCountry.js
函数名：	scoreMeet
参数：	数字,字符串
返回值：	字符串
方法签名：	function scoreMeet (numTeams, finishOrder)