

标题

(title.cpp)

【题目描述】

程序只有下列三种命令：

section: 创建新的一级标题，序号从 1 开始标记。

subsection: 创建新的二级标题，序号在每个一级标题的基础上从 1 开始标记。

subsubsection: 创建新的三级标题，序号在每个二级标题的基础上从 1 开始标记。

给定 n 组命令及标题名称，输出所有标题序号及其名称。

【输入格式】

第一行，包含一个正整数 n ，表示命令的数量。

接下来的 n 行，每行输入命令名称（**section**、**subsection**、**subsubsection** 之一）和标题名称。标题名称由不超过 20 个小写英文字母组成。

【输出格式】

共 n 行，表示所有标题序号及其名称。

【样例输入输出 1】

样例输入 (title.in)	样例输出 (title.out)
3 section zivotinje section boje section voce	1 zivotinje 2 boje 3 voce

【样例输入输出 2】

样例输入 (title.in)	样例输出 (title.out)
4 section zivotinje subsection macke subsection psi subsubsection mops	1 zivotinje 1.1 macke 1.2 psi 1.2.1 mops

【样例输入输出 3】

样例输入 (title.in)	样例输出 (title.out)
4 section zivotinje subsection psi section voce subsection ananas	1 zivotinje 1.1 psi 2 voce 2.1 ananas

【数据规模与约定】

Subtask 1 (10 pts) : $1 \leq n \leq 3$ 。

Subtask 2 (10 pts) : 只包含 **section** 命令。

Subtask 3 (10 pts) : 只包含 **section** 和 **subsection** 命令。

Subtask 4 (20 pts) : 无特殊限制。

对于 100% 的数据， $n \leq 100$ 。

比赛

(contest.cpp)

【题目描述】

有 N 位选手参加一个比赛。每个回合，第一名会得到 N 分，第二名会得到 $N - 1$ 分，以此类推，最后一名会得到 1 分。

现在第 i 位选手初始有 B_i 分。求多少选手经过一个回合，分数有机会变成所有选手中最

高的。

【输入格式】

输入共 $N + 1$ 行。

第一行，包含一个正整数 N ，表示选手总数。

接下来 N 行，每行包含一个整数 B_i ，表示第 i 位选手的初始分数。

【输出格式】

输出只有一行，包含一个整数，表示多少选手的分数有机会变成所有选手中最高的。

【样例输入输出 1】

样例输入 (contest.in)	样例输出 (contest.out)
3	3
8	
10	
9	

【样例输入输出 2】

样例输入 (contest.in)	样例输出 (contest.out)
5	4
15	
14	
15	
12	
14	

【数据规模与约定】

对于 100% 的数据， $3 \leq N \leq 3 \times 10^5$ ， $1 \leq B_i \leq 2 \times 10^6$ 。

登机

(boarding.cpp)

【题目描述】

由 m 人组成的某国代表团正前往澳大利亚参加 2013 年的 IOI。他们目前正在机场排队等候办理登机手续。有 n 个办理登机手续的服务台开放。一些官员的工作效率比其他官员高，所以服务台的运作速度不同。在第 k 个服务台，完成 1 名乘客的登机手续需要 t_k 秒，而我们的代表团成员恰好知道确切的数字。在开始时，所有的服务都做好接受下一名乘客的准备，而现在只允许我们的代表团成员排队。一个人只有在所有排在他前面的人都已经离开队列（开始办理登机手续，不一定要完成）时，才能占据一个可用的服务台开始办理登机手续。这时，这个人可以立即占据一张空闲的服务台（如果有的话），但也可以选择等待另一个办理登记手续更快的服务台变成空闲。

我们的代表团成员是计算机科学怪才，他们决定让他们所有人都尽快完成登机手续，你的任务是找到完成登机手续所需要花费的最短时间。

【输入格式】

输入共 $n + 1$ 行。

第一行两个整数 n 、 m ，分别表示服务台的个数和代表团的人数。

随后 n 行，每行一个整数 t_k ，表示第 k 个服务台办理一名乘客的登机手续需要花费的时间。

【输出格式】

输出仅一行一个整数，表示完成登机手续所需要花费的最短时间（单位为秒）。

【样例输入输出 1】

样例输入 (boarding.in)	样例输出 (boarding.out)
2 6 7 10	28

【样例输入输出 2】

样例输入 (boarding.in)	样例输出 (boarding.out)
7 10 3 8 3 6 9 2 4	8

【样例解释】

对于样例 1，有两个服务台，处理时间分别为 7 秒和 10 秒。在代表团的 6 个人中，前两个人立即占据了这 2 个服务台。在第 7 秒，第 1 张桌子变成空闲，第 3 个人占据了它。在第 10 秒，第 4 个人占据了第 2 个服务台。在第 14 秒，第 5 个人占据了第 1 个服务台。在第 20 秒，第 2 张桌子再次变成空闲，但第 6 个人决定再等 1 秒钟（至第 21 秒），让第 1 张桌子空出来，然后占据它。这样一来，签到只要花费 28 秒就能完成了。如果第 6 个人没有等待后到更快的服务台，签到将花费 30 秒。

【数据规模与约定】

本题一共 8 个测试点。各个测试点的数据范围如下表所示：

测试点编号	$n \leq$	$m \leq$	$t_k \leq$
1 ~ 5	10^5	3×10^5	10^9
6 ~ 8	10^5	10^9	10^9

货币系统 (money.cpp)

【题目描述】

在网友的国度中共有 n 种不同面额的货币，第 i 种货币的面额为 $a[i]$ ，你可以假设每一种货币都有无穷多张。为了方便，我们把货币种数为 n 、面额数组为 $a[1..n]$ 的货币系统记作 (n,a) 。

在一个完善的货币系统中，每一个非负整数的金额 x 都应该可以被表示出，即对每一个非负整数 x ，都存在 n 个非负整数 $t[i]$ 满足 $a[i] \times t[i]$ 的和为 x 。然而，在网友的国度中，货币系统可能是不完善的，即可能存在金额 x 不能被该货币系统表示出。例如在货币系统 $n=3$ ， $a=[2,5,9]$ 中，金额 1,3 就无法被表示出来。

两个货币系统 (n,a) 和 (m,b) 是等价的，当且仅当对于任意非负整数 x ，它要么均可以被两个货币系统表示，要么不能被其中任何一个表示。

现在网友们打算简化一下货币系统。他们希望找到一个货币系统 (m,b) ，满足 (m,b) 与原来的货币系统 (n,a) 等价，且 m 尽可能的小。他们希望你来协助完成这个艰巨的任务：找到最小的 m 。

【输入格式】

输入文件的第一行包含一个整数 T ，表示数据的组数。

接下来按照如下格式分别给出 T 组数据。

每组数据的第一行包含一个正整数 n 。

接下来一行包含 n 个由空格隔开的正整数 $a[i]$ 。

【输出格式】

输出文件共有 T 行，对于每组数据，输出一行一个正整数，表示所有与 (n,a) 等价的货币系统 (m,b) 中，最小的 m 。

【样例输入输出 1】

样例输入 (money.in)	样例输出 (money.out)
2	2
4	5
3 19 10 6	
5	
11 29 13 19 17	

【样例解释】

在第一组数据中，货币系统 $(2,[3,10])$ 和给出的货币系统 (n,a) 等价，并可以验证不存在 $m < 2$ 的等价的货币系统，因此答案为 2。在第二组数据中，可以验证不存在 $m < n$ 的等价的货币系统，因此答案为 5。

【数据规模与约定】

测试点	n	a _i	测试点	n	a _i
1	= 2	≤ 1000	11	≤ 13	≤ 16
2			12		
3			13		
4	= 3		14	≤ 25	≤ 40
5			15		
6			16		
7	= 4		17	≤ 100	≤ 25000
8			18		
9	= 5		19		
10			20		

对于 100% 的数据，满足 $1 \leq T \leq 20, n, a[i] \geq 1$ 。