CWNU 算协挑战赛

Round 2

2026 年 10 月 11 日

题目概况

	题目名	时间限制	空间限制
A	还差几个座位	1 s	256 MB
B	成对亮灯	1 s	256 MB
C	连号免单券	1 s	256 MB
D	安静的一侧	1 s	256 MB
E	打包的顺序	2 s	256 MB
F	双色通行证	2 s	256 MB
G	蒂娜的谜箱	1 s	256 MB

请勿使用生成式 AI 进行答题！



该试题册仅供参考，如有歧义请以评测系统题面为准。

Problem A. 还差几个座位

输入文件: standard input
输出文件: standard output
时间限制: 1 s
内存限制: 256 MB

算法协会准备举办一场聚餐，目前已经有 n 个人到场。场地可以提供足够多的桌子，每张桌子恰好有 k 个座位。

你可以继续邀请一些人参加聚餐，并安排所有人就座。你希望每个人恰好坐一个座位，并且每张使用中的桌子都恰好坐满。

请问至少还需要邀请多少人？你也可以不邀请任何人。

输入格式

第一行包含一个整数 t ($1 \leqslant t \leqslant 10^4$)，表示测试用例的数量。

接下来 t 行，每行包含两个整数 n, k ($1 \leqslant n, k \leqslant 10^9$)，分别表示已经到场的人数和每张桌子的座位数。

输出格式

对于每个测试用例，输出一行一个整数，表示至少还需要邀请的人数。

样例

standard input	standard output
5 10 4 12 4 3 8 1 1 1000000000 999999999	2 0 5 0 999999998

提示

第一组测试用例中，再邀请 2 人后共有 12 人，可以恰好坐满 3 张桌子。

第二组测试用例中，已有的 12 人已经可以恰好坐满 3 张桌子，因此无需继续邀请。

第三组测试用例中，再邀请 5 人后共有 8 人，可以恰好坐满 1 张桌子。

Problem B. 成对亮灯

输入文件: standard input
输出文件: standard output
时间限制: 1 s
内存限制: 256 MB

活动现场有 n 盏灯，它们的初始状态由一个长度为 n 的二进制字符串 s 表示：

- $s_i = 0$ 表示第 i 盏灯熄灭；
- $s_i = 1$ 表示第 i 盏灯点亮。

每次操作，你可以选择任意两盏不同的灯，同时翻转它们的状态：点亮的灯变为熄灭，熄灭的灯变为点亮。

你希望最终所有灯都处于点亮状态。请输出最少需要进行的操作次数。如果无法实现，输出 -1。

你可以进行零次操作。

输入格式

第一行包含一个整数 t ($1 \leqslant t \leqslant 10^4$)，表示测试用例的数量。

对于每个测试用例：

- 第一行包含一个整数 n ($1 \leqslant n \leqslant 2 \times 10^5$)，表示灯的数量；
- 第二行包含一个长度恰好为 n 的二进制字符串 s ($\forall s_i \in \{0, 1\}$)，表示灯的初始状态。

数据保证所有测试用例的 n 之和不超过 2×10^5。

输出格式

对于每个测试用例，输出一行一个整数，表示使所有灯点亮所需的最少操作次数。如果无法实现，输出 -1。

样例

standard input	standard output
6 5 10101 6 010010 4 1111 3 010 3 000 1 0	1 2 0 1 -1 -1

提示

第一组测试用例中，选择第 2 盏和第 4 盏灯进行一次操作，就可以使所有灯点亮。

第二组测试用例中，可以先选择第 1 盏和第 3 盏灯，再选择第 4 盏和第 6 盏灯，共进行两次操作。

第三组测试用例中，所有灯已经点亮，无需操作。

第四组测试用例中，选择第 1 盏和第 3 盏灯进行一次操作，就可以使所有灯点亮。选中的两盏灯不必相邻。

第六组测试用例中，只有一盏灯，无法选择两盏不同的灯进行操作。

Problem C. 连号免单券

输入文件: standard input
输出文件: standard output
时间限制: 1 s
内存限制: 256 MB

商店里有 n 件商品，按照给定顺序从 1 到 n 编号，第 i 件商品的价格为 a_i 元。

你需要购买全部 n 件商品。你有一张优惠券，可以选择两件编号相邻的商品，使这两件商品同时免单。

编号相邻指编号分别为 i 和 $i+1$，其中 $1 \leqslant i < n$。商品的编号和顺序都不能改变。

优惠券至多使用一次。请问购买全部商品最少需要支付多少钱?

输入格式

第一行包含一个整数 t $(1 \leqslant t \leqslant 10^4)$，表示测试用例的数量。

对于每个测试用例:

- 第一行包含一个整数 n $(2 \leqslant n \leqslant 2 \times 10^5)$，表示商品数量;
- 第二行包含 n 个整数 $a_1, a_2, \ldots, a_n$ $(1 \leqslant a_i \leqslant 10^9)$，表示各件商品的价格。

数据保证所有测试用例的 n 之和不超过 2×10^5。

输出格式

对于每个测试用例，输出一行一个整数，表示购买全部商品最少需要支付的金额。

样例

standard input
4 5 3 8 2 9 4 2 7 11 3 100 1 100 5 1000000000 1000000000 1000000000 1000000000 1000000000
standard output
13 0 100 3000000000

提示

第一组测试用例中，可以让第 4 件和第 5 件商品免单，需要支付 $3+8+2=13$ 元。

第二组测试用例中，两件商品可以同时免单，因此无需支付费用。

第三组测试用例中，第 1 件和第 3 件商品的编号不相邻，不能同时免单。可以让第 1 件和第 2 件商品免单，需要支付 100 元。

第四组测试用例中，任意两件编号相邻的商品免单后，都还需要支付另外三件商品的费用。

Problem D. 安静的一侧

输入文件: standard input
输出文件: standard output
时间限制: 1 s
内存限制: 256 MB

活动室里有一排 n 个座位。每个座位都有一个标记：0 表示安静区，1 表示交流区。从左到右，所有座位的标记组成一个长度为 n 的二进制字符串 s。

你希望调整这些标记，使得所有标记为 0 的座位，都位于所有标记为 1 的座位左侧。

每次操作，你可以选择一个座位，将它的标记从 0 改为 1，或从 1 改为 0。

请问最少需要进行多少次操作？所有座位都标记为 0，或者都标记为 1，也满足要求。你可以进行零次操作。

输入格式

第一行包含一个整数 t $(1 \leqslant t \leqslant 10^4)$，表示测试用例的数量。

对于每个测试用例：

- 第一行包含一个整数 n $(1 \leqslant n \leqslant 2 \times 10^5)$，表示座位数量；
- 第二行包含一个长度恰好为 n 的二进制字符串 s $(\forall s_i \in \{0, 1\})$，表示各个座位的初始标记。

数据保证所有测试用例的 n 之和不超过 2×10^5。

输出格式

对于每个测试用例，输出一行一个整数，表示使所有标记满足要求所需的最少操作次数。

样例

standard input	standard output
5 5 00111 6 010110 3 100 3 110 1 0	0 2 1 1 0

提示

第一组测试用例中，初始状态已经满足要求，无需操作。

第二组测试用例中，将第 2 个座位的标记改为 0，将第 6 个座位的标记改为 1，可以得到 000111，共进行两次操作。

第三组测试用例中，将第 1 个座位的标记改为 0，可以得到 000。

第四组测试用例中，将第 3 个座位的标记改为 1，可以得到 111

Problem E. 打包的顺序

输入文件:	standard input
输出文件:	standard output
时间限制:	2 s
内存限制:	256 MB

活动结束后，算法协会需要打包剩余的物资。

目前有 n 个包裹从左到右排成一行，第 i 个包裹的重量为 a_i。工作人员会不断合并包裹，直到只剩下一个包裹。

每次操作严格按照以下规则进行：

1. 在当前相邻的两个包裹中，选择重量之和最小的一对。
2. 如果有多对包裹满足条件，选择其中最靠左的一对。
3. 将它们合并为一个新包裹，其重量等于原来两个包裹的重量之和。新包裹放在它们原来的位置，其他包裹的相对顺序不变。
4. 本次操作消耗的时间，等于新包裹的重量。

请计算严格按照上述规则完成所有合并操作所消耗的总时间。如果最初只有一个包裹，则无需进行操作，总时间为 0。

输入格式

第一行包含一个整数 t（$1 \leqslant t \leqslant 10^4$），表示测试用例的数量。

对于每个测试用例：

- 第一行包含一个整数 n（$1 \leqslant n \leqslant 2 \times 10^5$），表示包裹的数量。
- 第二行包含 n 个整数 $a_1, a_2, \ldots, a_n$（$1 \leqslant a_i \leqslant 10^5$），依次表示从左到右各个包裹的重量。

保证所有测试用例的 n 之和不超过 2×10^5。

输出格式

对于每个测试用例，输出一行一个整数，表示完成所有合并操作所消耗的总时间。

样例

standard input	standard output
4 5 4 1 3 2 5 3 1 100 1 4 1 2 1 2 1 7	35 203 12 0

提示

第一组测试用例中，包裹的重量序列依次变化为：

$$[4,1,3,2,5] \to [4,4,2,5] \to [4,6,5] \to [10,5] \to [15].$$

四次操作分别消耗 $4,6,10,15$ 的时间，总时间为 35。

第二组测试用例中，两个重量为 1 的包裹并不相邻。第一次合并最左边的两个包裹，消耗 101 的时间；第二次合并剩余的两个包裹，消耗 102 的时间。总时间为 203。

第三组测试用例中，最初三对相邻包裹的重量之和均为 3，因此第一次必须合并最左边的一对。重量序列依次为 $[1,2,1,2]$、$[3,1,2]$、$[3,3]$、$[6]$，总时间为 $3+3+6=12$。

第四组测试用例中无需合并，总时间为 0。

Problem F. 双色通行证

输入文件:	standard input
输出文件:	standard output
时间限制:	2 s
内存限制:	256 MB

算法协会在一座有 n 个房间的建筑中举办寻路游戏。房间编号为 1 到 n，房间之间有 m 条双向通道。每条通道的颜色为红色或蓝色，分别用 0 和 1 表示。

你拥有一张可以变色的通行证。通行证在任意时刻也是红色或蓝色。移动规则如下：

1. 只有当通道的颜色与通行证的当前颜色相同时，才能通过这条通道。
2. 每次通过一条通道后，通行证立即变为另一种颜色。
3. 每次通过一条通道消耗 1 分钟。

游戏开始时，你位于房间 1，可以自由选择通行证的初始颜色。此后，通行证只能通过上述移动规则变色，停留不会使其变色。你可以重复经过房间和通道。

请计算从房间 1 到达房间 n 至少需要多少分钟。到达房间 n 时，通行证可以是任意颜色。如果无法到达，输出 -1。如果起点就是终点，则所需时间为 0。

输入格式

第一行包含一个整数 t（$1 \leqslant t \leqslant 10^4$），表示测试用例的数量。

对于每个测试用例：

- 第一行包含两个整数 n, m（$1 \leqslant n \leqslant 2 \times 10^5$，$0 \leqslant m \leqslant 2 \times 10^5$），分别表示房间的数量和通道的数量。
- 接下来 m 行，每行包含三个整数 u, v, c（$1 \leqslant u, v \leqslant n$，$u \neq v$，$c \in \{0, 1\}$），表示房间 u 和房间 v 之间有一条颜色为 c 的双向通道。

保证任意两个不同的房间之间至多有一条通道。所有测试用例的 n 之和不超过 2×10^5，m 之和不超过 2×10^5。

输出格式

对于每个测试用例，输出一行一个整数，表示从房间 1 到达房间 n 至少需要的分钟数。如果无法到达，输出 -1。

样例

standard input	standard output
3 5 5 1 2 0 2 5 0 1 3 1 3 4 0 4 2 1 3 2 1 2 0 2 3 0 1 0	4 -1 0

提示

第一组测试用例中，选择蓝色作为通行证的初始颜色，可以沿路线

$$1 \xrightarrow{1} 3 \xrightarrow{0} 4 \xrightarrow{1} 2 \xrightarrow{0} 5$$

到达终点。箭头上方的数字表示通道颜色，总共需要 4 分钟。

路线 $1 \to 2 \to 5$ 的两条通道均为红色。通过第一条通道后，通行证变为蓝色，无法立即通过第二条通道。

第二组测试用例中，两条通道均为红色，无法按照规则到达房间 3，答案为 -1。

第三组测试用例中，起点就是终点，答案为 0。

Problem G. 蒂娜的谜箱

输入文件: standard input
输出文件: standard output
时间限制: 1 s
内存限制: 256 MB

这是一道交互题。

蒂娜准备了 n 个方块，分别标有整数 $1, 2, \ldots, n$，以及从左到右编号为 $1, 2, \ldots, n$ 的 n 个格子。她已经为这些方块确定了一种正确的摆放顺序，并将它藏在了谜箱中。

正确顺序可以表示为一个排列 $p_1, p_2, \ldots, p_n$，其中 p_i 表示第 i 个格子中应当放置的方块编号。这个排列在交互开始前已经确定，在整个交互过程中不会改变。

你可以反复尝试摆放方块。每次尝试时，每个格子至多放置一个方块，同一个方块至多使用一次，也可以留下一些空格子。蒂娜只会告诉你：有多少个格子中放置了正确的方块。

你需要在不超过 $8n$ 次尝试后，确定并提交完整的正确顺序。最终提交不计入尝试次数。

输入格式

交互开始时，你的程序需要从标准输入读取一个整数 n $(2 \leqslant n \leqslant 100)$，表示方块和格子的数量。

单次程序运行只包含一局游戏，没有测试组数 t。隐藏排列 p 不会作为输入提供给选手程序。

此后，输入内容为交互器对询问命令的回复，具体格式见交互格式。最终提交后没有回复。

输出格式

你的程序需要按照交互格式向标准输出发送询问，并最终提交排列 p。

每条命令必须单独占一行，包含一个命令符和恰好 n 个整数。每次输出命令后，都必须换行并刷新输出缓冲区。

发送询问命令后，应读取交互器的回复；发送最终提交命令后，应立即结束程序，不再读取任何内容。

交互格式

你的程序可以发送以下两种命令。

1. **询问一次摆放方案**。

   输出一行：

   $$? \ a_1 \ a_2 \ \cdots \ a_n$$

   对于所有 $1 \leqslant i \leqslant n$，必须满足 $0 \leqslant a_i \leqslant n$。其中，$a_i = 0$ 表示第 i 个格子留空；$a_i > 0$ 表示在第 i 个格子中放置编号为 a_i 的方块。所有非零的 a_i 必须互不相同，0 可以重复出现。

   交互器回复一个整数

$$r = |\{i \mid 1 \leqslant i \leqslant n,\ a_i = p_i\}|,$$

表示正确放置的方块数量。空格子不计入正确数量。将所有格子留空也是一次合法询问，并计入询问次数。

你最多可以发送 $8n$ 次询问。即使某次询问的回复为 n，你仍然需要通过最终提交命令提交答案。

2. **提交最终答案**。

   输出一行：

   $$! \ b_1 \ b_2 \ \cdots \ b_n$$

   其中 $b_1, b_2, \ldots, b_n$ 必须是 $1, 2, \ldots, n$ 的一个排列。

   如果对于所有 $1 \leqslant i \leqslant n$ 都有 $b_i = p_i$，则答案正确；否则，答案错误。

   交互器不会对最终提交作出任何回复。输出最终答案、换行并刷新输出缓冲区后，你的程序必须立即结束，不应再尝试读取任何内容。

   最终答案只能提交一次，且不计入 $8n$ 次询问的限制。

如果你的程序发出了非法命令、询问次数超过限制，或提交了错误答案，交互器将回复 -1 并结束交互。**在任何时候，如果读取到 -1 或读取失败，你的程序都必须立即结束，不能继续输出**。

命令符与整数、相邻整数之间应使用空格或制表符分隔。行首、行末的空格或制表符不影响判定。不得输出空行、额外字段，或将一条命令拆成多行。

每次输出命令后必须刷新输出缓冲区。例如，使用 C++ 时可以调用 `cout.flush()`，或使用会同时换行并刷新的 `endl`。

样例

standard input	standard output
4 0 2 4	 ? 1 2 3 4 ? 0 1 0 2 ? 3 1 4 2 ! 3 1 4 2

提示

在样例中，$n = 4$，隐藏排列为 $(3, 1, 4, 2)$。隐藏排列仅在这里用于解释样例，不会由交互器发送给选手程序。

样例输入依次列出了交互器发送的内容，样例输出依次列出了选手程序发送的内容。**它们是双方交互记录的两个方向，不是普通的离线输入输出**。实际交互必须按以下顺序进行：

顺序	发送方	发送内容
1	交互器	`4`
2	选手程序	`? 1 2 3 4`
3	交互器	`0`
4	选手程序	`? 0 1 0 2`
5	交互器	`2`
6	选手程序	`? 3 1 4 2`
7	交互器	`4`
8	选手程序	`! 3 1 4 2`

第一次询问没有任何方块放在正确的位置，因此回复为 0。

第二次询问只在第 2 个格子和第 4 个格子中放置了方块，它们都正确，因此回复为 2。

第三次询问的四个位置都正确，因此回复为 4。选手程序随后发送最终提交命令，换行并刷新输出缓冲区后立即结束。交互器判定答案正确，不再发送回复。

样例只展示一种合法的交互过程，不表示能够用相同次数的询问解决所有隐藏排列。