



0 EDU API CALENDAR H0ME TOP CATALOG CONTESTS GYM PROBLEMSET GROUPS RATING HELP RAYAN 🟋

PROBLEMS SUBMIT CODE MY SUBMISSIONS STATUS HACKS ROOM STANDINGS CUSTOM INVOCATION

E. Keep the Sum

time limit per test: 2.5 seconds memory limit per test: 256 megabytes

You are given an integer k and an array a of length n, where each element satisfies $0 \le a_i \le k$ for all $1 \leq i \leq n$. You can perform the following operation on the array:

- Choose two distinct indices i and j ($1 \leq i, j \leq n$ and $i \neq j$) such that $a_i + a_j = k$.
- Select an integer x satisfying $-a_i \le x \le a_i$.
- Decrease a_i by x and increase a_j by x. In other words, update $a_i := a_i x$ and $a_i := a_i + x$.

Note that the constraints on x ensure that all elements of array a remain between 0 and kthroughout the operations.

Your task is to determine whether it is possible to make the array a non-decreasing* using the above operation. If it is possible, find a sequence of at most 3n operations that transforms the array into a non-decreasing one.

It can be proven that if it is possible to make the array non-decreasing using the above operation, there exists a solution that uses at most 3n operations.

 $\overline{^*$ An array a_1,a_2,\ldots,a_n is said to be non-decreasing if for all $1\leq i\leq n-1$, it holds that $a_i\leq a_{i+1}$.

Each test contains multiple test cases. The first line contains the number of test cases t ($1 < t < 10^4$). The description of the test cases follows.

The first line of each test case contains two integers, n and k ($4 \le n \le 2 \cdot 10^5$, $1 \le k \le 10^9$) — the length of the array a and the required sum for the operation.

The second line of each test case contains n integers a_1, a_2, \ldots, a_n ($0 \leq a_i \leq k$) — the elements of array a.

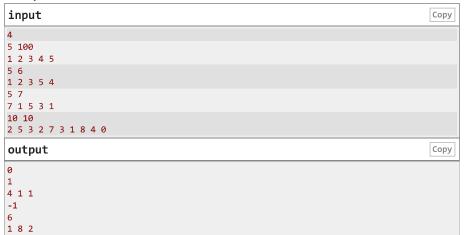
It is guaranteed that the sum of n over all test cases does not exceed $2 \cdot 10^5$.

For each test case, output -1 if it is not possible to make the array non-decreasing using the operation.

Otherwise, output the number of operations m ($0 \le m \le 3n$). On each of the next m lines, output three integers i, j, and x representing an operation where a_i is decreased by x and a_j is increased by x.

Note that you are **not** required to minimize the number of operations. If there are multiple solutions requiring at most 3n operations, you may output any.

Example



Codeforces Round 1019 (Div. 2)

Finished

Practice



→ Virtual participation

Virtual contest is a way to take part in past contest, as close as possible to participation on time. It is supported only ICPC mode for virtual contests. If you've seen these problems, a virtual contest is not for you solve these problems in the archive. If you just want to solve some problem from a contest, a virtual contest is not for you solve this problem in the archive. Never use someone else's code, read the tutorials or communicate with other person during a virtual contest.

Start virtual contest

→ Clone Contest to Mashup

You can clone this contest to a mashup.

Clone Contest



Choose file: Choose File No file chosen

Submit

→ Last submissions Submission Verdict Apr/22/2025 12:33 316659523 Accepted



×

→ Contest materials

Announcement (en)



Note

In the first test case, the array is already non-decreasing, so we do not need to perform any operations.

In the second test case, we can perform an operation with i=4, j=1, and x=1. a_4 decreases by 1 to become 5-1=4 while a_1 increases by 1 to become 1+1=2. After the operation, the array becomes [2,2,3,4,4], which is non-decreasing.

Note that there are other ways to make the array non-decreasing, all of which would be considered correct as long as they do not use more than $3 \cdot n = 15$ operations.

In the third test case, it is not possible to make the array non-decreasing. This is because there are no distinct pairs of indices i and j where $a_i+a_j=7$, so no operation can be done on the array.

In the fourth test case, the array is transformed as follows:

```
\begin{array}{l} \textbf{1.} \left[ \textbf{0,} 5, 3, 2, 7, 3, 1, \textbf{10}, 4, 0 \right] \\ \textbf{2.} \left[ 0, 5, \textbf{1}, 2, \textbf{9}, 3, 1, 10, 4, 0 \right] \\ \textbf{3.} \left[ 0, 5, 1, 2, \textbf{6}, 3, \textbf{4}, 10, 4, 0 \right] \\ \textbf{4.} \left[ 0, 5, 1, 2, \textbf{3}, 3, 4, 10, \textbf{7}, 0 \right] \\ \textbf{5.} \left[ 0, 5, 1, 2, 3, 3, 4, \textbf{5}, \textbf{7}, \textbf{5} \right] \\ \textbf{6.} \left[ 0, \textbf{1}, 1, 2, 3, 3, 4, \textbf{5}, \textbf{7}, \textbf{9} \right] \end{array}
```

Codeforces (c) Copyright 2010-2025 Mike Mirzayanov
The only programming contests Web 2.0 platform
Server time: Apr/22/2025 11:34:12^{UTC+2} (l1).
Desktop version, switch to mobile version.
Privacy Policy | Terms and Conditions

Supported by

