National Olympiad in Informatics

Finals Round 2

Important! Read the following:

Hidden Test Cases. Your solution will be checked by running it against one or more (usually several) hidden test cases. You will not have access to these cases, but a correct solution is expected to handle them correctly.

Strict Output Format. The output checker is **strict**. Follow these guidelines strictly:

- It is **space sensitive**. Do not output extra leading or trailing spaces. Do not output extra blank lines unless explicitly stated.
- It is **case sensitive**. So, for example, if the problem asks for the output in lowercase, follow it.
- Do not print any tabs. (No tabs will be required in the output.)
- Do not output anything else aside from what's asked for in the Output section. So, do not print things like "`Please enter t`".

Not following the output format strictly and exactly will likely result in the verdict "*Output isn't correct*".

Use Standard I/O. Do not read from, or write to, a file. You must read from the standard input and write to the standard output.

Submit Code Only. Only include **one** file when submitting: the source code (.cpp, .py, etc.) and nothing else.

No Java Package. For Java submissions, do not include a `package` line.

No Weird Filenames. Only use letters, digits and underscores in your filename. Do not use spaces or other special symbols.

Use Fast I/O. Many problems have large input file sizes, so use fast I/O. For example:

- In C/C++, use `scanf` and `printf`.
- In Python, use `sys.stdin.readline()`

Flush On Interactive Problems. On interactive problems, make sure to **flush** your output stream after printing.

- In C++, use `fflush(stdout);` or `cout << endl;`
- In Python, use `sys.stdout.flush()` or `print(flush=True)`
- For more details, including for other languages, ask a question/clarification through CMS.

Good luck and enjoy the contest! 😃

Contents

Notes

- Many problems have large input file sizes, so use fast I/O. For example:
  - In C/C++, use `scanf` and `printf`.
  - In Python, use `sys.stdin.readline()`
- On interactive problems, make sure to **flush** your output stream after printing.
  - In C++, use `fflush(stdout);` or `cout << endl;`
  - In Python, use `sys.stdout.flush()` or `print(flush=True)`
  - For more details, including for other languages, ask a question/clarification through CMS.

Good luck and enjoy the problems!

Problem C
Fast Threerier Transform

Hello everyone, welcome back to another episode of 2Blue2Brown - Tokyo Drift. As you recall, in last week's episode, we learned about how a fourier transform is a process which allows us to decompose an arbitrary signal into square waves (because squares have four *sides)[citation needed].*

This week, we're going to learn about a natural extension, the threerier *transform, which allows us to decompose a signal into* triangle waves*.*

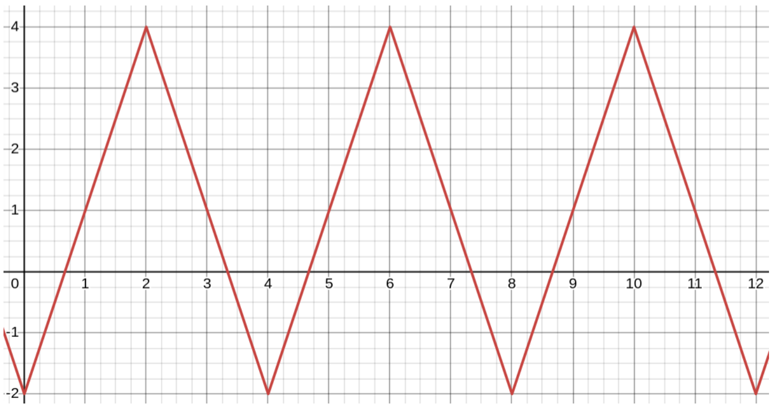
A **triangle wave** is a real-valued function described by three real-number values:

- b, the y-intercept
- m, the slope, which **must be nonzero**
- p, the half-period, which **must be positive**

It is a piecewise-linear function, and we define the triangle wave given by parameters (b, m, p) to be the unique *continuous* function T such that:

- $T(0) = b$
- If $0 \le x < p$, then $T(x)$ is a linear function[1] with a slope of m.
- If $p \le x < 2p$, then $T(x)$ is a linear function with a slope of $-m$.
- $T(x) = T(x - 2p)$ for all $x \in \mathbb{R}$.

Informally, it's a function whose plot has a shape that looks like this (here, $b = -2$, $m = 3$, and $p = 2$):



Note that the reason we disallowed $m = 0$ is because we wouldn't get these nice triangle shapes... we'd just get a line.

[1] Linear as in "functions whose plot is a line", like $y = mx + b$—nothing to do with the linear transformations we covered in our *Essence of Linear Algebra* series

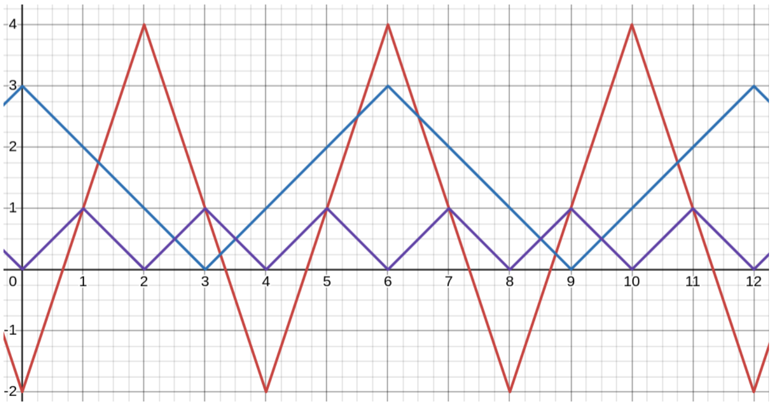
Your exercise for today is to reinvent the process known as the *discrete threerier transform*. Suppose we have n arbitrary data points, call them $a[0], a[1], a[2], \ldots,$ $a[n-1]$, representing an arbitrary signal.

What we need to do today is determine whether there exists a **non-empty** collection of triangle waves $T_1, T_2, \ldots, T_k$ such that:

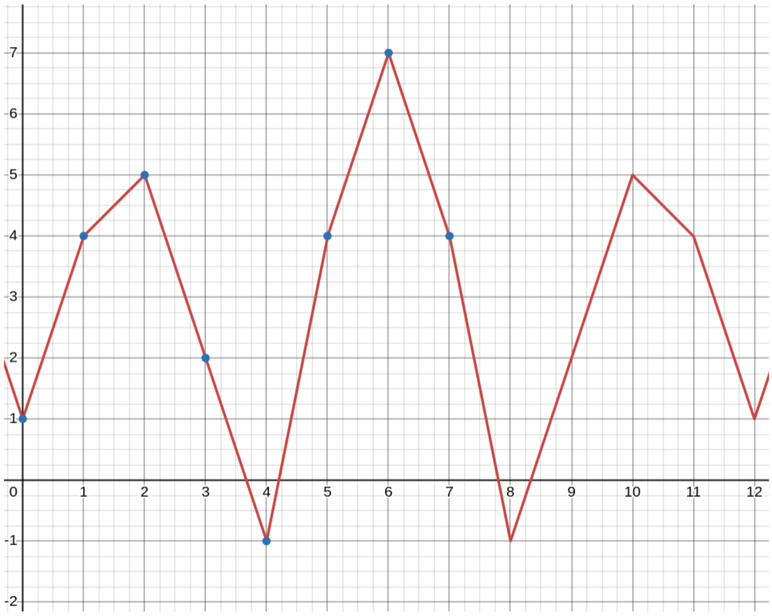
- $a[j] = T_1(j) + T_2(j) + \cdots + T_k(j)$, for all j from 0 to $n-1$.
- If T_i is the triangle wave described by the triple of integers (b_i, m_i, p_i), then b_i and m_i and p_i are all *integers*, for all i from 1 to k.

And, if a solution exists, you need to provide one of minimal size (i.e. the positive integer k, the number of triangle waves in the collection, must be as small as possible).

For example, if $a = [1, 4, 5, 2, -1, 4, 7, 4]$, we can decompose it into three triangle waves described by parameters $(-2, 3, 2)$, $(3, -1, 3)$, and $0, 1, 1$, illustrated below.



Summing up these three triangle waves gives us the following function, and we can see that the values it attains at inputs $x = 0, 1, 2, \ldots, n-1$ match the desired values from our given signal.



Input Format

The first line of input contains the single integer n.

The second line contains the n space-separated integers $a[0], a[1], \ldots, a[n-1]$.

Output Format

If a discrete threerier transform does not exist for the given input, output a line containing the integer -1.

If one does exist, instead output a line containing a positive integer k, the size of the smallest non-empty collection of triangle waves which satisfies the conditions described above.

Then, output k lines, each containing three space-separated integers, describing the triangle waves in the collection. The ith line should contain the three parameters b_i, m_i, and p_i. Each of these values should be between -10^{18} and 10^{18}, and also $m_i \neq 0$ and $p_i > 0$ must hold. It can be shown that if a solution exists, then one exists with these constraints.

If multiple answers exist, any will be accepted.

Constraints

For all subtasks
$1 \leq n \leq 2 \times 10^5$ $\lvert a[i] \rvert \leq 998244353$ for all i.

Subtask	Points	Constraints
1	**15**	If an answer exists, it uses at most 1 triangle wave.
2	**25**	If an answer exists, it uses at most 2 triangle waves.
3	**11**	$\lvert a[i] \rvert \leq 5$ for all i
4	**25**	$n \leq 2000$
5	**24**	No further constraints.

Sample I/O

<table>
<tr><th>Input 1</th><th>Output 1</th></tr>
<tr><td><pre>8
1 4 5 2 -1 4 7 4</pre></td><td><pre>3
-2 3 2
3 -1 3
0 1 1</pre></td></tr>
</table>