Triple Peaks

The Cordillera Oriental is a mountain range in the Andes that stretches across Bolivia. It consists of a sequence of N mountain peaks, numbered from 0 to $N-1$. The **height** of peak i ($0 \leq i < N$) is $H[i]$, which is an integer between 1 and $N-1$, inclusive.

For any two peaks i and j where $0 \leq i < j < N$, the **distance** between them is defined as $d(i,j) = j - i$.

According to ancient Inca legends, a triple of peaks is **mythical** if it has the following special property: the heights of the three peaks **match** their pairwise distances **ignoring the order**.

Formally, a triple of indices (i, j, k) is mythical if

- $0 \leq i < j < k < N$, and
- the heights $(H[i], H[j], H[k])$ match the pairwise distances $(d(i,j), d(i,k), d(j,k))$ ignoring the order. For example, for indices $0, 1, 2$ the pairwise distances are $(1, 2, 1)$, so the heights $(H[0], H[1], H[2]) = (1, 1, 2)$, $(H[0], H[1], H[2]) = (1, 2, 1)$, and $(H[0], H[1], H[2]) = (2, 1, 1)$ all match them, but the heights $(H[0], H[1], H[2]) = (1, 2, 2)$ do not match them.

This problem consists of two parts, with each subtask associated with either **Part I** or **Part II**. You may solve the subtasks in any order. In particular, you are **not** required to complete all of Part I before attempting Part II.

Part I

Given a description of the mountain range, your task is to count the number of mythical triples.

Implementation Details

You should implement the following procedure.

```
long long count_triples(std::vector<int> H)
```

- H: array of length N, representing the heights of the peaks.
- This procedure is called exactly once for each test case.

The procedure should return an integer T, the number of mythical triples in the mountain range.

Constraints

- $3 \le N \le 200\,000$
- $1 \le H[i] \le N - 1$ for each i such that $0 \le i < N$.

Subtasks

Part I is worth a total of 70 points.

Subtask	Score	Additional Constraints
1	8	$N \le 100$
2	6	$H[i] \le 10$ for each i such that $0 \le i < N$.
3	10	$N \le 2000$
4	11	The heights are non-decreasing. That is, $H[i-1] \le H[i]$ for each i such that $1 \le i < N$.
5	16	$N \le 50\,000$
6	19	No additional constraints.

Example

Consider the following call.

```
count_triples([4, 1, 4, 3, 2, 6, 1])
```

There are 3 mythical triples in the mountain range:

- For $(i, j, k) = (1, 3, 4)$, the heights $(1, 3, 2)$ match the pairwise distances $(2, 3, 1)$.
- For $(i, j, k) = (2, 3, 6)$, the heights $(4, 3, 1)$ match the pairwise distances $(1, 4, 3)$.
- For $(i, j, k) = (3, 4, 6)$, the heights $(3, 2, 1)$ match the pairwise distances $(1, 3, 2)$.

Hence, the procedure should return 3.

Note that the indices $(0, 2, 4)$ do not form a mythical triple, as the heights $(4, 4, 2)$ do not match the pairwise distances $(2, 4, 2)$.

Part II

Your task is to construct mountain ranges with many mythical triples. This part consists of 6 **output-only** subtasks with **partial scoring**.

In each subtask, you are given two positive integers M and K, and you should construct a mountain range with **at most** M peaks. If your solution contains **at least** K mythical triples, you will receive the full score for this subtask. Otherwise, your score will be proportional to the number of mythical triples your solution contains.

Note that your solution must consist of a valid mountain range. Specifically, suppose your solution has N peaks (N must satisfy $3 \leq N \leq M$). Then, the height of peak i ($0 \leq i < N$), denoted by $H[i]$, must be an integer between 1 and $N - 1$, inclusive.

Implementation Details

There are two methods to submit your solution, and you may use either one for each subtask:

- **Output file**
- **Procedure call**

To submit your solution via an **output file**, create and submit a text file in the following format:

```
N
H[0] H[1] ... H[N-1]
```

To submit your solution via a **procedure call**, you should implement the following procedure.

```
std::vector<int> construct_range(int M, int K)
```

- M: the maximum number of peaks.
- K: the desired number of mythical triples.
- This procedure is called exactly once for each subtask.

The procedure should return an array H of length N, representing the heights of the peaks.

Subtasks and Scoring

Part II is worth a total of 30 points. For each subtask, the values of M and K are fixed and given in the following table:

Subtask	Score	M	K
7	5	20	30
8	5	500	2000
9	5	5000	$50\,000$
10	5	$30\,000$	$700\,000$
11	5	$100\,000$	$2\,000\,000$
12	5	$200\,000$	$12\,000\,000$

For each subtask, if your solution does not form a valid mountain range, your score will be 0 (reported as `Output isn't correct` in CMS).

Otherwise, let T denote the number of mythical triples in your solution. Then, your score for the subtask is:

$$5 \cdot \min\left(1, \frac{T}{K}\right).$$

Sample Grader

Parts I and II use the same sample grader program, with the distinction between the two parts determined by the first line of the input.

Input format for Part I:

```
1
N
H[0] H[1] ... H[N-1]
```

Output format for Part I:

```
T
```

Input format for Part II:

```
2
M K
```

Output format for Part II:

```
N
H[0] H[1] ... H[N-1]
```

Note that the output of the sample grader matches the required format for the output file in Part II.