Wimpy’s diet (diet)

Wimpy is always hungry, but his doctor put him on a diet: he’s only allowed to eat sandwiches in a strictly decreasing order of weight at any given meal.

His favorite restaurant, however, only serves sandwiches in a fixed order, so Wimpy has to decide which ones to pick.

You are to help him: write a program that removes sandwiches from a menu of \( N \) sandwiches so that the remaining sandwiches have the maximum possible total weight and are served in a strictly decreasing order of weight.

**Example:**
Given a menu of 8 sandwiches:

\[
389 \ 207 \ 155 \ 300 \ 299 \ 170 \ 158 \ 65
\]

We can remove the sandwiches 207 and 155 to obtain a decreasing sequence:

\[
389 \ 300 \ 299 \ 170 \ 158 \ 65
\]

Of maximum total weight 1381.

**Implementation**

You should submit a single file, with either a `.c`, `.cpp`, `.java` or `.py` extension. Your program must read input data from `stdin` and write the output data into `stdout`.

`stdin` consists of only one line:

- Line 1: The integer \( N \), the number of sandwiches on a menu.
- Line 2: \( N \) integers space separated, the weight of the sandwiches.

`stdout` consists of only two lines:

- Line 1: The number of sandwiches in the solution.
- Line 2: The weights of the remaining sandwiches of the menu.

**Constraints**

- \( 1 \leq N \leq 10.000 \).
- \( 1 \leq W[i] \leq 10.000 \) for all \( 0 \leq i < N \).
- No two sandwiches have the same weight.

**Scoring**

Your program will be tested on a number of testcases grouped in subtasks. In order to obtain the score associated to a subtask, you need to correctly solve all testcases of which it is formed.
• Subtask 1  [40 points]: \( N \leq 1.000. \)
• Subtask 2  [60 points]: \( N \leq 10.000. \)

**Examples**

<table>
<thead>
<tr>
<th>stdin</th>
<th>stdout</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 389 207 155 300 299 170 158 65</td>
<td>6 389 300 299 170 158 65</td>
</tr>
<tr>
<td>4 16 93 107 224</td>
<td>1 224</td>
</tr>
</tbody>
</table>