The Peak-End Rule

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

The **weight** of a sequence $[v_1, \ldots, v_m]$ of length m is defined as follows:

1. If $m = 1$, then the weight is v_1.
2. If $m > 1$, then the weight is the sum of the last number in the sequence and the maximum value among the preceding elements, i.e., $\max_{i=1}^{m-1} v_i + v_m$.

You are given a sequence $a_1, a_2, \ldots, a_n$. You wish to partition it into at most k non-empty subsequences such that the sum of their **weights** is maximized.

A "subsequence" is a sequence obtained by deleting 0 or more elements from the original sequence without changing the relative order of the remaining elements.

Each element of the original sequence must belong to exactly one subsequence after partitioning.

Input

The first line contains two integers n and k ($n \geq 2$, $1 \leq k \leq n \leq 2 \times 10^5$).

The second line contains n integers a_i ($-10^9 \leq a_i \leq 10^9$).

Output

Output two lines. The first line contains a single integer, representing the maximum possible sum of the weights of the subsequences obtained from the partition.

The second line contains n integers id_i ($1 \leq id_i \leq k$), where elements with the same id_i are assigned to the same subsequence. Since we do not require exactly k non-empty subsequences, it is not necessary that every integer between 1 and k appears among the id_i values.

Examples

standard input	standard output
7 2 6 3 7 5 6 4 5	24 1 2 2 1 1 2 2
5 2 -5 -4 -3 -2 -1	-3 1 1 1 1 1

Note

In the first example, one possible partition is into two subsequences: $[a_1, a_4, a_5] = [6, 5, 6]$ and $[a_2, a_3, a_6, a_7] = [3, 7, 4, 5]$. The sum of their weights is $12 + 12 = 24$.