

Problem B: Mad Scientist

Source file: mad.{c, cpp, java}

Input file: mad.in

A mad scientist performed a series of experiments, each having n phases. During each phase, a measurement was taken, resulting in a positive integer of magnitude at most k . The scientist knew that an individual experiment was designed in a way such that its measurements were monotonically increasing, that is, each measurement would be at least as big as all that precede it. For example, here is a sequence of measurements for one such experiment with $n=13$ and $k=6$:

1, 1, 2, 2, 2, 2, 2, 4, 5, 5, 5, 5, 6

It was also the case that n was to be larger than k , and so there were typically many repeated values in the measurement sequence. Being mad, the scientist chose a somewhat unusual way to record the data. Rather than record each of n measurements, the scientist recorded a sequence P of k values defined as follows. For $1 \leq j \leq k$, $P(j)$ denoted the number of phases having a measurement of j or less. For example, the original measurements from the above experiment were recorded as the P -sequence:

2, 7, 7, 8, 12, 13

as there were two measurements less than or equal to 1, seven measurements less than or equal to 2, seven measurement less than or equal to 3, and so on.

Unfortunately, the scientist eventually went insane, leaving behind a notebook of these P -sequences for a series of experiments. Your job is to write a program that recovers the original measurements for the experiments.

Input: The input contains a series of P -sequences, one per line. Each line starts with the integer k , which is the length of the P -sequence. Following that are the k values of the P -sequence. The end of the input will be designated with a line containing the number 0. All of the original experiments were designed with $1 \leq k < n \leq 26$.

Output: For each P -sequence, you are to output one line containing the original experiment measurements separated by spaces.

Example Input:	Example Output:
6 2 7 7 8 12 13 1 4 3 4 4 5 3 0 4 5 5 2 2 4 7 7 0	1 1 2 2 2 2 2 4 5 5 5 5 6 1 1 1 1 1 1 1 1 3 2 2 2 2 3 1 1 3 3 4 4 4