Problem K
Keep it Cool
Time limit: 2 seconds



Picture by Fitness First via Vimeo, cc by

As the workload of the semester is ramping up you get the task of refilling the fridge in the lab with soda. The fridge has s slots, each with a capacity for d bottles of soda, and you have n new soda bottles to add to the fridge. The sodas currently in the fridge are all nice and cold, but the new ones are not and need to be cooled in the fridge for a while until they are ready to drink.

You can only refill the fridge from the front, so in an ideal world, you would first take out all the sodas currently in the fridge, then put in the n new ones, and then put the old and cold sodas in front of the new ones. But in an ideal world you would also not have two exams and a homework deadline coming. You are simply way too busy to do all this work.

Instead, you are going to just put the new bottles in the front of the fridge and hope for the best. However, you can still to be clever about which slots to put the new sodas in. Each time a student comes for a soda, they will take one from the front of a uniformly random non-empty slot in the fridge. You decide to add the new bottles to the fridge so as to maximize the probability that all the next m students getting a soda from the fridge will get a cold one.

Input

The first line of input contains four integers n, m, s and d ($1 \leq n, m, s, d \leq 100$), the number of new soda bottles, number of students to optimize for, number of slots in the fridge, and capacity of each slot, respectively. Then follows a line containing s integers $c_1, \ldots, c_s$ ($0 \leq c_i \leq d$ for each i), where c_i is the number of soda bottles currently in slot i of the fridge.

You may assume that there is free space for all the n new bottles in the fridge.

Output

If there is a chance that all the next m students will get a cold bottle, then output s integers describing a refill scheme for the n soda bottles that maximizes the probability of this happening. The i^{th} of these s integers indicates how many of the new bottles are placed in the front of slot i in the fridge. If there are multiple optimal refill schemes, output any one of them. Otherwise, if it is impossible for all the next m students to get a cold soda, output "`impossible`" instead.

Sample Input 1	Sample Output 1
`5 3 3 4` `0 1 4`	`2 3 0`

Sample Input 2	Sample Output 2
`2 7 6 4` `0 1 2 2 0 1`	`impossible`

This page is intentionally left blank.