# Supper Hammer Tap Game

Imagine that there exists a beautiful virtual world which is divided into land and underground spaces, in one far dark day, those stick-like animals in the underground space decide to rise up to conquest all the land, this is very bad, because in consequences – destroy the balance, they can destroy all that beautiful world. You are being to be sent there to stop them and make order. The only way to fight them is tapping on theirs heads using a hammer. You have to fight them all in an optimal way.
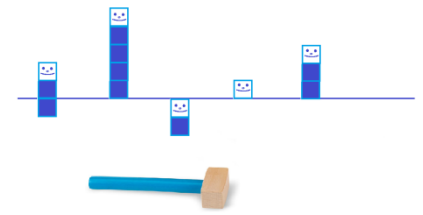
Here is the mission in details:

There are n underground animals denoted as $S = \{a_1, a_2, ..., a_n\}$. The time system in used is the whole numbers of time units, begins from 1. Animals can raise up at different time, we know that the animal a_i will appeared at r_i and as soon as it appeared on the land you can start to tap it. By tapping on the animal, it will go back to the underground by 1 unit of height. We also know that the height of each animal a_i is p_i units, consequently you have to tap a_i, in general, p_i times to make it back completely, taps are not required to be continuous.

With the limit of only one hammer, at one time you can tap only one animal among those are on the land, any one of them. If all animals are in the underground, you can't tap them, but you can relax. One more, such as we are in a digital world, your actions will be translated into integer numbers, a relax will be translated to 0, and a tap on a_i – an integer number i, where i is in the range $[1..n]$.

Let c_i be the time of the last tap on a_i, which makes it went completely to the underground, we call c_i the disappearence time of the animal a_i. Such as the longer those animals exists on the land, the more damage they could made, so it's required to tap them in an optimal sequence, which minimizes the sum of their disappearence times, unless they will destroy the world.

Input: Input data is provided in a *input.txt* text file, which contains three lines. The first line contains only one integer number – n – the number of animals, the second line contains n number $r_1...r_n$, the third line contains n number $p_1...p_n$, numbers are seperated by one space.

Output: Write your outputs to *output.txt* text file in two lines. The first line contains one integer number – the minimum value for the sum of disappearence times, the second line contains the corresponding sequence of numbers, which describe your optimal sequence of actions, numbers should be seperated by one space.

An Input and output example:

input.txt	output.txt
2	9
2 2	0 1 1 2 2 2
2 3	

The input.txt means: n = 2; r_1 = 2, r_2 = 2; p_1 = 2; p_2 = 3. The output.txt means the sum of disappearence times is 9, at time = 1 you relaxs, after that you taps a_1 twice, so c_1 = 3, and then you taps a_2 three times, so c_2 = 6 and $c_1 + c_2$ = 9.

You can tap those animals in diffirent ways, e.g., 0 1 2 1 2 2, for this sequence c_1 = 4, c_2 = 6, and the sum of disappearance times is 4 + 6 = 10, so it is not an optimal sequence.

There will be many tests, and your outputs will be verified to be correct or wrong, e.g.,

The following 3 outputs	8	8	8
are wrong:	2 2 2 1 1	0 2 2 1 1	0 1 1 2 2 2
	a_2 not appeared	*a_2 is not completed*	*sum is wrong*

=== End of task description ===