PROBLEM 4

There are n cups of different sizes, and three trays A, B, and C. Initially the cups are piled up upside down on the trays, at most one pile on each tray, such that the smallest cup (among the cups on the tray) is at the bottom, on which the second smallest, on which the third smallest, and so on.

For example, Fig. 1 shows the case where $n = 5$ and 2 cups are piled up on tray A, nothing on tray B, and 3 on tray C.

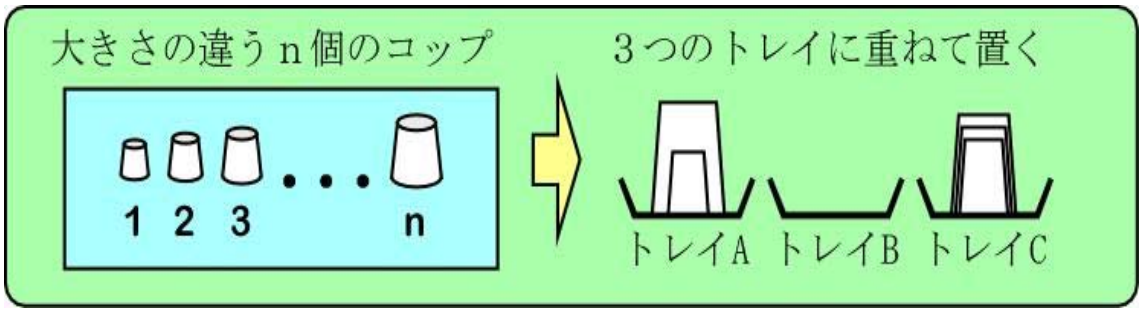

Figure 1: An initial configuration, where $n = 4$.

Given an initial configuration, find the smallest number of steps to move all the cups so that they are piled up either on tray A only or on tray C only, where the moves must obey the following three rules.

Rule 1 At each step, only the topmost cup on some tray (which is the largest one on the tray) can be moved to be piled up on the topmost cup on another tray.

Rule 2 Do not pile up any smaller cup on any other cup.

Rule 3 Neither direct move "from tray A to C" nor "from C to A" is allowed. Thus, a direct move must be one of "from A to B", "from B to A", "from B to C", and "from C to B."

Your task is to write a program which, given an initial configuration of n cups and an integer m, determines whether or not all the cups can be moved to be piled up on tray A whithin m steps. If it is possible, then the program should output the smallest number of steps to reach the final configuration, and -1 otherwise.

INPUT

The first line of each input file contains 2 integers n, the number of cups ($1 \leq n \leq 15$) and m, an upper bound of the number of steps permitted ($1 \leq m \leq 15000000$), separeted by a space character. Each of the 2nd, 3rd, and 4th lines contains an integer k, followed by a sequence of integers $c_1, \cdots, c_k$ taken from $\{1, 2, \cdots, n\}$ sorted in the ascending order. The k on the 2nd, 3rd, and 4th lines represent the numbers of cups piled up initially on trays A, B, and C, respectively. The $c_1, \cdots, c_k$ on the 2nd, 3rd, and 4th lines represent the (sizes of) cups piled up on trays A, B, and C, respectively.

OUTPUT

Each output file must contain a single line containing either the required number of steps or -1 followed by the Return code.

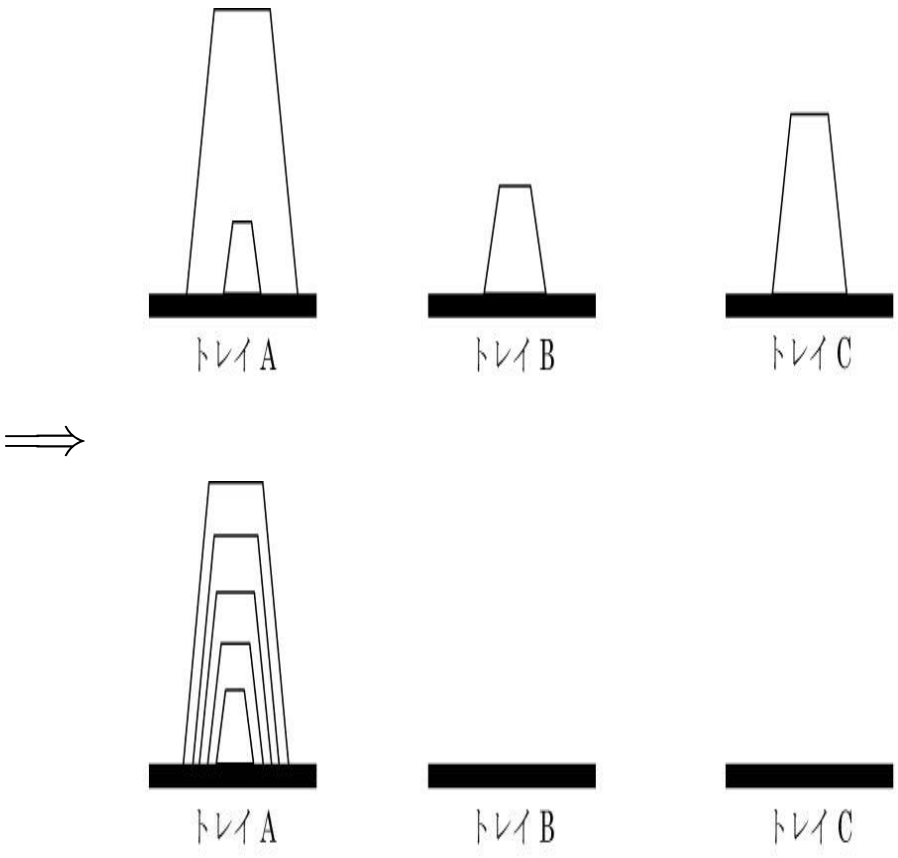

Figure 2: All of the $n\,(=4$ in this case) cups are moved to be piled up on tray A.

EXAMPLE

Example input 1	Example input 2	Example input 3	Example input 4
3 10 0 1 1 2 2 3	4 20 2 1 2 1 3 3	2 5 2 1 2 0 0	3 3 0 1 1 2 2 3
Example output 1	**Example output 2**	**Example output 3**	**Example output 4**
9	3	0	-1