Crossing River

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

Panda is a boatman who operates a boat on a river. The boat can carry at most one passenger at a time.

One day, there are n people who want to cross from the left bank of the river to the right bank. Their arrival times at the left riverbank are $a_1, a_2, \ldots, a_n$. There are also m people who want to cross from the right bank to the left bank. Their arrival times at the right riverbank are $b_1, b_2, \ldots, b_m$. A passenger can only board the boat at or after their arrival time.

The boat takes exactly k units of time to cross the river, whether it has a passenger or not. Panda can choose the boat's starting position (left or right bank). The goal is to find a crossing schedule that minimizes the time when the very last of the $n + m$ people reaches the opposite bank destination. You should also help Panda determine the complete crossing schedule.

Input

The first line contains three integers n, m, k ($1 \le n, m \le 10^5$, $1 \le k \le 10^9$), denoting the number of people starting on the left, the number of people starting on the right, and the time for a single crossing.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$), which are the arrival times for the people on the left bank.

The third line contains m integers $b_1, b_2, \ldots, b_m$ ($1 \le b_i \le 10^9$), which are the arrival times for the people on the right bank.

Output

The first line should contain one integer T, representing the minimum possible time the last passenger reaches their destination.

The following $n + m$ lines should describe the crossing schedule **in chronological order**. The i-th line contains three integers t_i, tp_i, id_i ($0 \le t_i \le T$, $tp_i \in \{0, 1\}$), indicating that at time t_i, the passenger with index id_i from the left bank ($tp_i = 0$) or the right bank ($tp_i = 1$) gets on the boat. When $tp_i = 0$, you must ensure that $1 \le id_i \le n$ and $t_i \ge a_{id_i}$; when $tp_i = 1$, you must ensure that $1 \le id_i \le m$ and $t_i \ge b_{id_i}$. Moreover, you must ensure that $t_1 < t_2 < \cdots < t_{n+m}$ and $T = \max_i(t_i + k)$.

For a valid schedule, every one of the $n+m$ people must successfully board the boat and reach the opposite bank. The time between two consecutive boarding times must be at least k, that is, $t_i - t_{i-1} \ge k$ for all $1 < i \le n + m$. If two consecutive people board from the same bank ($tp_i = tp_{i-1}$), the time between their boarding must be at least $2k$, that is, $t_i - t_{i-1} \ge 2k$.

Example

`standard input`	`standard output`

`5 5 2`
`2 1 13 19 11`
`12 18 19 7 8`
	`25`
	`5 0 2`
	`7 1 4`
	`9 0 1`
	`11 1 5`
	`13 0 5`
	`15 1 1`
	`17 0 3`
	`19 1 2`
	`21 0 4`
	`23 1 3`