

1277 – Looking for a Subsequence

Given a sequence of integers $S = \{a_1, a_2, a_3 \dots a_n\}$ and an integer m , you have to find a subsequence of S containing m elements, $P = \{b_1, b_2, b_3 \dots b_m\}$ such that $(b_1 < b_2 < \dots < b_m)$. If there are several subsequences possible then you should find the one where b_1 is leftmost. If there is still a tie, then check for the one where b_2 is leftmost and so on.

For example, let the sequence be, 8 7 5 6 5 1 2 7 and $m = 2$. Then (1, 2), (5, 6), (5, 7), (1, 7) ... etc can be solutions. But we are looking for (5, 6).

Input

Input starts with an integer $T (\leq 12)$, denoting the number of test cases.

Each case starts with a line containing two integers $n (1 \leq n \leq 10^5)$ and $q (1 \leq q \leq 10)$ where q denotes the total number of queries. The next line contains n space separated integers denoting $a_1, a_2 \dots a_n$ respectively. Each of the next q lines contains an integer $m (1 \leq m \leq n)$. You can assume that $-10^8 \leq a_i \leq 10^8$.

Output

For each case, print the case number in a line. Then for each query print a single line containing the desired subsequence or 'Impossible' if there is no such subsequence. Insert a single space between two integers of a subsequence.

Sample Input	Output for Sample Input
3	Case 1:
6 3	Impossible
3 4 1 2 3 6	1 2 3 6
6	3 4
4	Case 2:
2	2 4 6
6 2	Impossible
2 4 6 1 3 5	Case 3:
3	8
4	5 6
8 5	5 6 7
8 7 5 6 5 1 2 7	Impossible
1	Impossible
2	
3	
4	
5	

Note

Dataset is huge, use faster I/O methods.