Monks' Game of Cards

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

Moudud, Shantanu, and Sumit are playing a game with a deck of n cards. In this deck, each card is marked with a number written on it. Sumit will shuffle the deck of cards, and then Shantanu will cut the deck at a certain position. Moudud will have to guess the number written on the card on top of the deck after the shuffle and cut.

To prevent the game from becoming a boring game of chance, Sumit will tell Moudud his shuffle order. Before every round, Sumit will tell Moudud that he will shuffle the deck k times and Shantanu will tell Moudud his cut position p. After every round, the deck will be rearranged to its original order.

Now Moudud wants your help to win. Can you help him find the number written on the card on top of the deck?

Definitions:

A deck of cards is represented by an array a of size n, where a_i denotes the number written on the i^{th} card. The position of the cards is numbered from top to bottom and a_1 is the number written on the card at the top. The i^{th} card can also be referred to as the card at position i

A shuffle order is a permutation of size n (array of size n consisting of all integers from 1 to n), denoted as s, where each element s_i represents the original position of the card that should occupy the position i after shuffling. Applying a shuffle with this order rearranges the deck such that the card at position s_i before the shuffle moves to position i. Formally, $a'_i = a_{s_i}$, where a_i is the number written on the i^{th} card before the shuffle, a'_i is the number written on the i^{th} card after the shuffle and s is the shuffle order.

In the cut operation at a specified position p, the deck is split at the position p. The deck is divided into two parts: the first part consists of cards from position 1 to position p, and the second part consists of cards from position $p+1$ to the end. After cutting, the two parts are swapped (the second part is brought before the first), resulting in a new arrangement of cards. Formally, cutting a deck a_1, a_2, ..., a_n transforms it into a_{p+1}, a_{p+2}, ..., a_n, a_1, a_2, ..., a_p.

Input

The first line of the input contains a single positive integer t ($1 \le t \le 10^5$) — the number of test cases. Then t test cases follow.

The first line of every test case contains two space-separated integers $n(2 \le n \le 2\times10^5)$ and $r(1 \le r \le 10^5)$ — the number of cards in the deck and the number of rounds they will play the game.

The second line of every test case contains n space-separated integers a_1, a_2, ..., a_n — the array representing the deck of cards, where a_i ($1 \le a_i \le 10^9$) denotes the number written on the i^{th} card.

The third line of every test case contains n space-separated integers s_1, s_2, ..., s_n that contains all integers from 1 to n — Sumit's shuffle order.

The last r lines of every test case contain two space-separated integers $k\,(0 \le k \le 10^{18})$ and $p\,(1 \le p < n)$ — the number of times Sumit will shuffle the deck and the position where Shantanu will cut.

The sum of $n \times r$ will not exceed 2×10^5 over all test cases.

Output

For each test case, output r lines with one integer per line, representing the number written on the card on top of the deck in each round.

Example

standard input	standard output

2
5 3
10 20 30 40 50
3 5 4 1 2
2 4
0 1
50 2
2 1
3 5
2 1
1 1
	50
	20
	10
	3

Note

In the first round of the first test case, the initial deck is $10, 20, 30, 40, 50$.

After the first shuffle, it becomes $30, 50, 40, 10, 20$.

After the second shuffle, it becomes $40, 20, 10, 30, 50$.

After the cut, it becomes $50, 40, 20, 10, 30$.