Sensors

Input file:	`standard input`
Output file:	`standard output`
Time limit:	5 seconds
Memory limit:	1024 megabytes

There are n red balls arranged in a row, numbered from 0 to $(n-1)$ (both inclusive) from left to right. We are going to perform n operations, where the i-th operation will color the a_i-th ball to blue. After all operations all the balls will become blue.

There are m sensors, numbered from 1 to m (both inclusive), monitoring the color of the balls. The i-th sensor will become active, if there is exactly one red ball among all the balls numbered from l_i to r_i (both inclusive); Otherwise the sensor remains inactive.

Determine which sensors are active after each operation.

More precisely, let's say there are k_i active sensors after the i-th operation and their indices are $s_{i,1}, s_{i,2}, \cdots, s_{i,k_i}$. For each $0 \le i \le n$, output $v_i = \sum_{j=1}^{k_i} s_{i,j}^2$. Specifically, define $v_0 = \sum_{j=1}^{k_0} s_{0,j}^2$, where k_0 is the number of active sensors before the first operation, and the indices of the active sensors are $s_{0,1}, s_{0,2}, \cdots, s_{0,k_0}$.

Input

There are multiple test cases. The first line of the input contains an integer T indicating the number of test cases. For each test case:

The first line contains two integers n and m ($1 \le n, m \le 5 \times 10^5$) indicating the number of balls and the number of sensors.

For the following m lines, the i-th line contains two integers l_i and r_i ($0 \le l_i \le r_i < n$) indicating the detection range of the i-th sensor.

The next line contains n integers $a'_1, a'_2, \cdots, a'_n$ ($0 \le a'_i < n$) where a'_i indicates the **encoded** i-th operation. The real value of a_i is equal to $(a'_i + v_{i-1}) \bmod n$, where v_{i-1} is the answer after the $(i-1)$-th operation, defined in the description above. With the encoded operations, you're forced to calculate the answer to each operation before processing the next one. It's guaranteed that each a_i is distinct after decoding.

It's guaranteed that neither the sum of n nor the sum of m of all test cases will exceed 5×10^5.

Output

For each test case, output one line containing $(n+1)$ integers $v_0, v_1, \cdots, v_n$ separated by a space. The meaning of v_i is defined in the description above.

Example

standard input	standard output
3 5 4 2 4 2 3 3 3 0 2 3 2 4 2 0 2 1 1 1 1 0 2 1 0 1 0 0	9 13 29 17 16 0 1 1 0 0 1 0

Note

For the first sample test case:

- Before the first operation, only sensor 3 is active, so $v_0 = 3^2 = 9$.
- For the 1-st operation, the real $a_1 = (3 + 9) \bmod 5 = 2$. After this operation, sensors 2 and 3 are active, so $v_1 = 2^2 + 3^2 = 13$.
- For the 2-nd operation, the real $a_2 = (2 + 13) \bmod 5 = 0$. After this operation, sensors 2, 3 and 4 are active, so $v_2 = 2^2 + 3^2 + 4^2 = 29$.
- For the 3-rd operation, the real $a_3 = (4 + 29) \bmod 5 = 3$. After this operation, sensors 1 and 4 are active, so $v_3 = 1^2 + 4^2 = 17$.
- For the 4-th operation, the real $a_4 = (2 + 17) \bmod 5 = 4$. After this operation, only sensor 4 is active, so $v_4 = 4^2 = 16$.
- For the 5-th operation, the real $a_5 = (0 + 16) \bmod 5 = 1$. After this operation, no sensor is active, so $v_5 = 0$.