

D: Fantastic Problem

Andrew, the world-renowned problem writer, has finally decided to retire. However, he first wishes to put out one final masterpiece to seal his legacy and amaze competitors the world over. As you're his favorite pupil, he's asked you to help him proofread it before he submits it to the ICFP (International Committee for Fantastic Problems).

The problem proves to be quite the involved number theory affair, but you get together a solution eventually…only to find that it doesn't pass his data. After wasting hours debugging your code, you realize that it's correct – the data is invalid! Andrew's advanced age seems to have really caught up with him.

In his problem, you're given a sequence of ***n*** integers, $V_{1..n}$, with the important guarantee that within every interval of **k** consecutive integers ($V_i..V_{i+k-1}$, for some starting index **i**), any two integers will be coprime (two integers are coprime if they share no common factors besides 1). Note However, this condition is not met in Andrew's data, causing your program to crash!

In an attempt to help your beloved mentor out, you count how many size-**k** intervals don't satisfy his problem statement's promise. Unfortunately, your work is not done yet. Having difficulty fixing his data, Andrew makes ***m*** sequential updates, Andrew makes ***m*** sequential updates, where each update involves selecting some position ***a*** in the number sequence, and changing its value to some value ***b***. After each change, he wants to know how many invalid size-**k** intervals his new sequence contains. As if that wasn't enough, after the m^{th} update, Andrew decides that the data is good enough, and wants you to solve his problem with the resulting sequence of integers! His problem statement is as follows:

Given a sequence of integers, compute their sum.

The Input

There will be several test cases in the input. Each test case will begin with a line with three integers, ***n*** ($1 \le n \le 100{,}000$), ***k*** ($1 \le k \le n$) and ***m*** ($1 \le m \le 100{,}000$), where ***n*** is the size of Andrew's list, ***k*** is the size of the consecutive intervals of interest, and ***m*** is the number of changes Andrew makes. On each of the next ***n*** lines will be a single integer ***v*** ($1 \le v \le 100{,}000$), which is a value in Andrew's input. The ***v***'s will be in the order that they appear in Andrew's list. On each the following ***m*** lines will be a pair of integers ***a*** ($1 \le a \le n$) and ***b*** ($1 \le b \le 100{,}000$), indicating that Andrew has changed value V_a to be ***b***. The input will end with a line with three 0s.

The Output

For each test case, output ***m***+2 integers. The first integer should be the number of size-**k** intervals in Andrew's original list which fail the pairwise-coprime constraint. Each of the next ***m*** integers should be the number of size-**k** intervals that fail the constraint after each of Andrew's ***m*** changes, in order. The final integer should be the sum of the numbers in the final list. Output each integer on its own line, with no spaces. Do not print any blank lines between outputs.

Sample Input	Sample Output
6 3 4 7 2 3 4 5 6 4 3 5 9 4 10 6 11 0 0 0	2 3 3 3 2 42