Knights of Night

Input file:	`standard input`
Output file:	`standard output`
Time limit:	7 seconds
Memory limit:	1024 megabytes

On the sixteenth night of the phantasmagoria, the knights gather around, prepared for a fight.

Two groups of knights are set to fight, each group consists of n knights. The i-th knight in the first group has a strength of a_i, and the j-th knight in the second group has a strength of b_j.

Your task is to organize the fights between the knights for tonight. Each fight involves one knight from each group, and each knight can participate in at most one fight. The outcome of the fights is not important—what truly matters is how wild, or lunatic, the fights become.

The *lunaticus* of a fight is defined as the sum of the strengths of the two knights involved. However, there's a special condition: if the lunaticus of a fight reaches 998244353, it overflows and resets to zero. Specifically, the fight between the i-th knight from the first group and the j-th knight from the second group has a lunaticus of $(a_i + b_j) \bmod 998244353$.

The special condition applies only to the lunaticus of individual fights. When calculating the sum of lunaticus for multiple fights, the sum of lunaticus can exceed 998244353.

Since the number of fights has not been decided, you would like to calculate that, for each integer x from 1 to k, what is the maximum possible sum of lunaticus for exactly x fights held tonight. In addition, there are m pairs of specific fights that cannot be organized. Please maximize the sum of lunaticus while avoiding these m restrictions.

Input

The first line contains three integers n, m, k ($1 \le n \le 10^5, 0 \le m \le 3 \times 10^5, 1 \le k \le \min(n, 200)$), representing the number of knights per group, the number of additional pairs of fights that cannot be organized, and the number of questions you need to answer.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($0 \le a_i < 998244353$), representing the knights' strength in the first group.

The third line contains n integers $b_1, b_2, \ldots, b_n$ ($0 \le b_i < 998244353$), representing the knights' strength in the second group.

Followed by m lines, the i-th of which contains two integers u_i, v_i ($1 \le u_i, v_i \le n$), representing that you cannot organize a fight by the pair of the u_i-th knight in the first group, and the v_i-th knight in the second group. It is guaranteed that all given pairs are distinct.

Output

Output k numbers in one line, the i-th of which represents the maximum possible sum of lunaticus for exactly i fights held tonight. For the i-th number, if it is impossible to organize i fights, output -1 instead.

Examples

standard input	standard output
3 4 3 10 998244352 5 998244352 8 6 1 2 1 3 2 2 3 3	998244351 998244364 27
6 10 5 749208241 448597025 773867529 808779198 281439035 850615248 796547348 327547103 1 4 5 3 3 4 2 1 5 6 4 3 6 2 6 3 2 6 2 4	882168541 1667618296 2328768389 2900938913 3354309143 554621438 19807774 765641981 702406342
1 1 1 0 0 1 1	-1