Problem A. Three Arrays

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	256 mebibytes

You are given three arrays: a containing n_a elements, b containing n_b elements and c containing n_c elements. These arrays are sorted in non-decreasing order: that is, for every i such that $1 \le i < n_a$ we have $a_i \le a_{i+1}$, for every j such that $1 \le j < n_b$ we have $b_j \le b_{j+1}$, and for every k such that $1 \le k < n_c$ we have $c_k \le c_{k+1}$.

Your task is to calculate the number of triples (i, j, k) such that $|a_i - b_j| \le d$, $|a_i - c_k| \le d$, and $|b_j - c_k| \le d$.

Input

The input contains one or more test cases. Each test case consists of four lines.

The first line of each test case contains four integers: d, n_a, n_b, and n_c ($1 \le d \le 10^9$, $1 \le n_a, n_b, n_c \le 5 \cdot 10^5$).

The second line contains n_a integers $a_1, a_2, \ldots, a_{n_a}$: the array a ($-10^9 \le a_i \le 10^9$).

The third line contains n_b integers $b_1, b_2, \ldots, b_{n_b}$: the array b ($-10^9 \le b_i \le 10^9$).

The fourth line contains n_c integers $c_1, c_2, \ldots, c_{n_c}$: the array c ($-10^9 \le c_i \le 10^9$).

All arrays are sorted in non-decreasing order. The total sum of n_a over all testcases does not exceed $5 \cdot 10^5$. The total sum of n_b over all testcases does not exceed $5 \cdot 10^5$. The total sum of all n_c over all testcases does not exceed $5 \cdot 10^5$. The test cases just follow one another without any special separators.

Output

For each test case, print a single integer: the number of triples (i, j, k) such that $|a_i - b_j| \le d$, $|a_i - c_k| \le d$, and $|b_j - c_k| \le d$.

Example

standard input	standard output
1 3 3 3 1 2 3 1 2 3 1 2 3 1 6 6 6 1 1 2 2 3 3 2 2 3 3 4 4 3 3 4 4 5 5	15 56