Substring Pairs

Input file:	*standard input*
Output file:	*standard output*
Time limit:	7 seconds
Memory limit:	512 MB

You are given two strings S and T, both consisting only of decimal digits, and an integer K.

For a string X, let $\text{ms}(X)$ denote the multiset of its characters. For two multisets A and B of digits, the size of their intersection is

$$|A \cap B| = \sum_{c=0}^{9} \min\big(\text{cnt}_c(A),\ \text{cnt}_c(B)\big),$$

where $\text{cnt}_c(\text{X})$ is the number of occurrences of the digit c in the multiset X.

Count the pairs (A, B) where A is a contiguous substring of S, B is a contiguous substring of T, and

$$\big|\, \text{ms}(A) \cap \text{ms}(B) \,\big| \geq K.$$

Two pairs are considered different if they differ in the position of A inside S or in the position of B inside T. In other words, you must count the quadruples (i, j, k, l) with $1 \leq i \leq j \leq |S|$ and $1 \leq k \leq l \leq |T|$ such that the substrings $S[i \ldots j]$ and $T[k \ldots l]$ satisfy the condition above; equal substrings occurring at different positions are counted separately.

Input

The first line contains the string S.

The second line contains the string T.

The third line contains a single integer K.

Both strings consist only of characters `0`–`9`.

$1 \leq |S|, |T| \leq 3000$

$1 \leq K \leq 5$

Output

Output a single integer: the number of substring pairs.

Example

standard input	standard output
121 12 1	15
1231 3213 3	9

Note

In the first example, S has 6 substrings (by position) and T has 3, giving 18 pairs in total; the 3 pairs that do not share any digit are (`"1"`, `"2"`) for each of the two occurrences of `'1'` in S, and (`"2"`, `"1"`). All the remaining 15 pairs are counted. Note that the substring `1` of S is counted twice, once for each of its two positions.