Problem F. Just Shuffle the Input

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3 seconds
Memory limit:	512 mebibytes

A *permutation* p of size n is a sequence of n pairwise distinct numbers from 1 to n. We denote the i-th of them by $p(i)$. By $p^k(i)$ we denote $\underbrace{p(p(p(\ldots(p(i))\ldots)))}_{k \text{ times}}$. A permutation is called *cyclic* if the minimal positive k for which $p^k(1) = 1$ equals n.

You are given a string s of size n, a string t of size m and a cyclic permutation p of size m. You want t to be a substring of s. To do this, you may apply the *shuffle* operation zero or more times. The shuffle operation consists of replacing t with t', such that the i-th letter of t equals the $p(i)$-th letter of t' for each i from 1 to m.

Please find out if it is possible obtain a substring of s. If it is possible, find the minimum number of shuffles required.

Recall that a string a is a substring of s if there exists some l such that $1 \le l \le |s| - |a| + 1$ and $s_{l+i-1} = a_i$ for every i from 1 to $|a|$.

Input

The first line of input contains two integers n and m ($1 \le m \le n \le 200\,000$). The second line contains m integers $p(1), \ldots, p(m)$: the permutation you can apply. The next two lines contain string s of length n and string t of length m, respectively. Both strings consist of lowercase English letters.

It is guaranteed that the permutation in the input is cyclic.

Output

If it is impossible to obtain a substring of s from t, output -1. Otherwise print the minimum number of shuffles needed to obtain a substring of s.

Examples

standard input	standard output
3 2 2 1 aba ba	0
7 4 3 4 2 1 dcabadc abcd	1