Task 4: `Digits`

Jayden is a math nerd who is obsessed with numbers! His favourite number is an m-digit string x. Ziv gives him n other m-digit strings $v[1], v[2], \ldots, v[n]$. All digits in these strings (including x) range from 0 to $k-1$, where k is a given integer ($2 \leq k \leq 10$). Let $v[i][j]$ denote the j-th digit of $v[i]$ from the left.

As Jayden loves his favorite number x so much, he wishes to turn all the n numbers that Ziv gave him into the number x using a number transformation machine. An operation on $v[i]$ works as follows:

- Choose two integers l and r where $1 \leq l \leq r \leq m$.
- For every $l \leq j \leq r$, set the value of $v[i][j]$ to $(v[i][j] + a[j]) \bmod k$.

a is a given array of length m. $p \bmod q$ denotes the remainder of dividing p by q (for example, $5 \bmod 2 = 1$). The cost of this operation is $c[l] + c[r]$ dollars (in particular, if $l = r$, the cost is $c[l] + c[l]$), where c is a given array of length m. Refer to the sample test cases for more details.

For each $v[i]$, help Jayden solve the following problem **independently**:

- What is the minimum total cost needed (in dollars) to transform $v[i]$ into the number x using any number of operations?

If it is impossible to transform $v[i]$ into x, output -1 instead.

Input Format

Your program must read from standard input.

The first line of input contains three space separated integers n, m, and k.

The second line of input contains m space-separated integers $a[1], a[2], \ldots, a[m]$.

The third line of input contains m space-separated integers, $c[1], c[2], \ldots, c[m]$.

The fourth line of input contains one integer x.

The next n lines of input each contain one integer. The i-th of these lines contains $v[i]$.

Output Format

Your program must print to standard output.

The output should contain n lines, each containing one integer. The i-th of these lines should contain the minimum total cost needed to transform $v[i]$ into x. If this is impossible, output -1 instead.

Subtasks

For all test cases, the input will satisfy the following bounds:

- $1 \leq n \leq 200\,000$
- $1 \leq m \leq 5$
- $2 \leq k \leq 10$
- $1 \leq a[i] \leq k - 1$ for all $1 \leq i \leq m$
- $1 \leq c[i] \leq 10^9$ for all $1 \leq i \leq m$
- $x, v[1], v[2], \ldots, v[n]$ are all m-digit strings, where each digit ranges from 0 to $k - 1$ inclusive. **They may contain leading zeros.**

Your program will be tested on input instances that satisfy the following restrictions:

Subtask	Score	Additional Constraints
0	0	Sample test cases
1	5	$m = 1$ and $a[i] = 1$ for all $1 \leq i \leq m$
2	13	$m = 2$ and $a[i] = 1$ for all $1 \leq i \leq m$
3	10	$k = 2$ and $c[1] = c[2] = \cdots = c[m]$
4	16	$c[1] = c[2] = \cdots = c[m]$
5	24	$n \leq 20$
6	32	No additional constraints

Sample Test Case 1

This test case is valid for subtasks 5 and 6.

Input	Output
6 3 8 1 2 3 3 1 4 676 356 431 676 767 133 715	16 42 0 -1 25 37

Sample Test Case 1 Explanation

Jayden's favourite number is $x = 676$.

Consider $v[1] = 356$. The following sequence of 3 operations can transform 356 into 676:

$$\underline{35}6 \xrightarrow{l=1,\ r=2} \underline{4}76 \xrightarrow{l=1,\ r=1} \underline{5}76 \xrightarrow{l=1,\ r=1} 676$$

1. $l = 1,\ r = 2$: The 1-st and 2-nd digits become $(3+1) \bmod 8 = 4$ and $(5+2) \bmod 8 = 7$ respectively. This costs $c[1] + c[2] = 3 + 1 = 4$ dollars.
2. $l = 1,\ r = 1$: The 1-st digit becomes $(4+1) \bmod 8 = 5$. This costs $c[1]+c[1] = 3+3 = 6$ dollars.
3. $l = 1,\ r = 1$: The 1-st digit becomes $(5 + 1) \bmod 8 = 6$. This costs 6 dollars.

The total cost of the three operations is $4 + 6 + 6 = 16$ dollars. It can be shown that there is no other sequence of operations that incurs a lower total cost.

For $v[3] = 676$, no operations have to be made as the number is already equal to x. Hence, the minimum total cost is 0 dollars.

For $v[4] = 767$, it can be shown that there is no sequence of operations that can transform 767 into 676. Hence, output -1.

Sample Test Case 2

This test case is valid for subtasks 3, 4, 5, and 6.

Input	Output
3 4 2 1 1 1 1 1 1 1 1 1001 1110 1100 0110	2 4 2

Sample Test Case 3

This test case is valid for subtasks 1, 4, 5, and 6.

Input	Output
1 1 10 1 67 6 7	1206

Sample Test Case 4

This test case is valid for subtasks 2, 5, and 6.

Input	Output
1 2 10 1 1 1 1000000000 24 83	1000000007