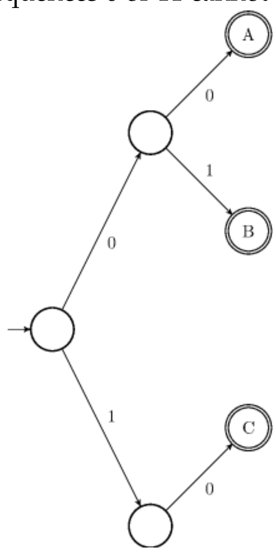
Unique Coding

A `binary code` is a translation of letters from an alphabet to binary `codewords`. A binary code can be used for instance to send messages over a wire. An example of a binary code is the representation of letters A, B, and C by 2-bit binary codewords ($A \mapsto 00$, $B \mapsto 01$, and $C \mapsto 10$). One can visualize this binary code as a rooted tree (see figure) whose leaves correspond to the letters and the codewords are obtained by concatenating the bits on the path from the root to a leaf. To decode the binary sequence 100001, we traverse the tree from the root and end up at the leaf corresponding to C after the first two bits. Then we start over from the root and decode A from the next two bits. Finally, we decode B from the last two bits. Hence, the unique string that can be decoded from the binary sequence 100001 is CAB. In general, since no codeword is a prefix of another one, it is always possible to uniquely decode any binary sequence to a string of letters (or determine that the binary sequence cannot be decoded – for instance, the binary sequences 0 or 11 cannot be decoded).



An alternative binary code for the letters A, B, and C is as follows: $A \mapsto 1$, $B \mapsto 10$, and $C \mapsto 100$. This code is not prefix-free (e.g., the codeword for A is a prefix of any other codeword). Nevertheless, any binary sequence can still be uniquely decoded by counting how many zeros follow a one. As an example, the binary sequence 100110 can be decoded to CAB. The binary sequence 1000 cannot be decoded.

Now consider the well-known `Morse code`. The binary sequence 01 (a dot followed by a dash) can be decoded to a single letter A or two letters ET. Furthermore, the binary sequence 00000 (five dots) can be decoded to two letters HE or SI.

Mouse Stofl has invented his own binary code for sending messages. Unfortunately, he didn't verify whether the code is uniquely decodable. Attempting to fix this issue, Mouse Stofl has decided to send the length of the original message separately. He believes that this approach will resolve the issue of unique decodability.

The Morse code example shows that providing the original message's length can help (e.g., for the binary sequence 01), but need not help (e.g., for the binary sequence 00000). Your task is given a binary code and a binary sequence, check whether it is uniquely decodable and if so, compute its unique decoding.

Input

The first line contains three space-separated integers N, M, and K ($1 \leq K \leq 8$), denoting the length of the original message, the length of the encoded message, and the size of the binary code's alphabet, respectively. The binary code's alphabet consists of the first K upper-case letters of the English alphabet. The second line contains M bits without spaces — the encoded message. The i-th of the following K lines contains the codeword for the i-th letter from the binary code's alphabet (without spaces).

You may assume that the (non-empty) codewords are pairwise distinct and the length of no codeword exceeds 8. You may further assume that the original message consists only of letters from the binary code's alphabet.

Output

If the encoded message is not decodable, output "`not decodable`" (without quotes). If the encoded message is decodable, but not uniquely decodable, output "`not uniquely decodable`" (without quotes). Otherwise, output the unique decoded message (using upper-case letters).

Limits

- Subtask 1: $1 \leq N \leq 10$, $1 \leq M \leq 20$, $1 \leq K \leq 4$.
- Subtask 2: $1 \leq N \leq 1\,000$, $1 \leq M \leq 10\,000$, $1 \leq K \leq 8$.

Sample

Input	Output
3 6 3 100001 00 01 10	CAB

Input	Output
3 6 3 100110 1 10 100	CAB

Input	Output
1 4 3 1000 1 10 100	not decodable

Input	Output

`2 1 4`
`0`
`0`
`00`
`000`
`0000`
	`not decodable`

Input	Output

`2 5 4`
`00000`
`0`
`00`
`000`
`0000`
	`not uniquely decodable`