

# Dictionary Subsequences

You have a dictionary of strings, and you want to perform some queries on the strings. In particular, you're given a single string  $T$ , and for each word  $W$  in the dictionary, you want to determine if  $W$  is a subsequence of  $T$ . A string  $B$  is a subsequence of a string  $C$  if you can remove zero or more of  $C$ 's letters to form a string equal to  $B$  (but the order of remaining letters may not be rearranged).

Each word  $W$  in the dictionary will be described in the input as a run length encoded (RLE) string. That is,  $W$  will be described by several pairs of data values, where each pair of data values consists of a positive integer  $K$  with no leading zeros and a letter  $L$ . A data pair with values  $K$  and  $L$  represents a string with  $K$  occurrences of the character  $L$ . To get the uncompressed string, we concatenate all strings represented by the data pairs. For example, the RLE string  $2A1B5C12A$  represents the string  $AABCCCCAAAAAAAAAAAA$ .

## Input

The first line of the input contains a positive integer  $C$  ( $0 < C < 10$ ), the number of test cases to follow. Each case begins with a line containing a positive integer  $D$  ( $0 < D < 10000$ ) representing the number of dictionary words and a string  $T$  with length between 1 and 10000.  $D$  lines follow, with each line containing a string with length between 1 and 200 in RLE format, which represents a dictionary word with uncompressed length between 1 and 10000. All uncompressed strings ( $T$  and dictionary words) will consist only of uppercase letters ('A'-Z').

## Output

Output for each case consists of several lines. There should be one line per dictionary word  $W$  (in the order of appearance in input) that will say either "YES" if  $W$  is a subsequence of  $T$ , or "NO" otherwise. Print a blank line after each test case.

## Example

### Input:

```
1
5 EFFERVESCENT
2E
1E1F1V1C1E
1E2F1C1R
1S2E
1P1E2F
```

### Output:

```
YES
YES
NO
YES
NO
```