

J: Voting Duels

Source file name: `voting.c`, `voting.cpp`, or `voting.java`

In order to choose their president, the congressmen of X must follow strict election bylaws. If there are n candidates, the election must be accomplished by means of rounds, each facing two of the candidates in a *voting duel*. The loser of a round is eliminated and the winner remains in the competition. If there is a tie, both candidates remain (although the corresponding duel will not be repeated). At the end, every pair of remaining candidates should have had their duel and, since several candidates could be finally tied, in this case they all are elected presidents.

Congressmen are grouped in disjoint blocs of different sizes. Each bloc has its particular opinion and its members sort the candidates according to their particular preferences. If a round must decide among candidates a and b , each bloc gives everyone of its votes for the candidate they more like in its preference list.

To illustrate the process, suppose there are four candidates, a , b , c and d and that congressmen are grouped as follows (for each bloc, size in brackets and preferences in descending order):

- Bloc 1: [17] $c a d b$
- Bloc 2: [32] $a b d c$
- Bloc 3: [34] $d b c a$
- Bloc 4: [17] $b a c d$

Denoting by $x:y$ a round to vote x against y , the result of an election accomplished by rounds $a:b$, $b:d$ and $c:d$ would have d as winner. To see that, note that

- $a:b$ results in 17 (Bloc 1) plus 32 (Bloc 2) votes for a , and 34 (Bloc 3) plus 17 votes for b (Bloc 4), i.e, b wins (49 against 51);
- $b:d$ results in 32 (Bloc 2) plus 17 (Bloc 4) votes for b , and 17 (Bloc 1) plus 34 votes for d (Bloc 3), i.e, d wins (49 against 51);
- $c:d$ results in 17 (Bloc 1) plus 17 (Bloc 4) votes for c , and 32 (Bloc 2) plus 34 votes for d (Bloc 3), i.e, d wins (34 against 66).

But it should be clear that if the rounds order were $b:d$, $c:d$, $a:d$ the winner would be a .

Some congressmen have observed this situation and want to analyze it more deeply. They want to know whether, for a particular bloc grouping with given preference lists is it possible that one specific candidate could be the winner if an appropriate rounds order is chosen. Could you help them?

Input

In order to identify candidates in the input, when in a case there are n candidates to be considered, each one of them is denoted with one of the first n lower case English alphabet letters.

The input consists of several test cases. The first line of a case contains two integer numbers n ($1 \leq n \leq 26$) and b ($1 \leq b \leq 100$), and one English alphabet letter, indicating, respectively, the number of candidates, the number of blocs, and the identifier that represents the designated candidate (for whom it should be decided if he/she can be president through an appropriate voting order).

Then there are b lines, so that the i -th of these lines indicates the size and the preferences of the i -th bloc, with an integer number s indicating the size of the bloc ($1 \leq s \leq 50$), followed by a word that is a permutation of the first n letters of the English alphabet (lower case). This last word defines the preference voting order for the i -th bloc: leftmost identifiers in this word represent candidates that are preferred to right most ones.

You may assume that the identifier read in the first line is one of the first n English alphabet letters.

Data in lines is left justified and separated by one blank.

The input must be read from standard input.

Output

For each test case, print one line with exactly one letter 'Y' or 'N', indicating if there is or there is not a way to define the order of the voting duels so that the designated candidate (that one in the first input line) is elected president (eventually, not alone).

The output must be written to standard output.

Sample Input	Sample Output
4 4 c	Y
17 cadb	N
32 abdc	
34 dbca	
17 bacd	
3 2 a	
12 cba	
10 bca	