

# Symphony in c++ major

Input file:            **standard input**  
Output file:          **standard output**  
Time limit:           4 seconds  
Memory limit:        256 megabytes

Making new songs is hard, so we asked an AI (ChatDoReMi) to create some new songs for us. The results... are pretty horrible! It turns out that LLMs are totally not-suited for composing music. Who could have guessed? But maybe we can use our own creativity and salvage something out of it.

A string is called **symphonic** if it can be produced by concatenating some (possibly empty) sequence of notes, where each note is one of “do”, “re”, “mi”, “fa”, “so”, “la”, “ti” (without the single quotes, of course). So, for example:

- “dodo”, “doremifasolatido”, “redososomiso”, “doremitiladoremisofadoredo” and “” (the empty string) are all symphonic;
- “dod”, “r”, “soda”, and “icpcamritapuri” are **not** symphonic.

Given a “reference string”  $s$  that was provided to us by ChatDoReMi, you must be able to measure how far from being symphonic certain substrings of  $s$  are — and actually, you should also be able to handle some updates, too, since we might ask ChatDoReMi to refine its response.

Formally, you must be able to process  $q$  queries, each of which is one of two kinds:

- “ $? i j$ ” means that you must consider the substring that starts at position  $i$  and ends at position  $j$ , inclusive (this is 1-indexed). What is the minimum number of letters that must be deleted from this substring in order to make it symphonic? Note that each query is an independent and hypothetical question — no letters actually get deleted.
- “ $\# i j new$ ” means that you must replace the  $i$ -th through  $j$ -th characters of  $s$  (also inclusive and 1-indexed) with the string  $new$ ; it is guaranteed that  $|new| = j - i + 1$ .

## Input

The first line of input contains two space-separated integers  $n$  and  $q$ , where  $n$  is the length of  $s$ .

The second line contains the string  $s$ .

Then  $q$  lines follow, describing the queries. Each query consists of a single line that is of the form “ $? i j$ ” or “ $\# i j new$ ”, as described above.

- $1 \leq n \leq 5 \cdot 10^5$
- $1 \leq q \leq 3 \cdot 10^5$
- For each query,  $1 \leq i \leq j \leq n$
- The total length of  $new$  across all “ $\#$ ” queries is  $\leq 10^5$
- Each string consists of lowercase English letters.

## Output

For each “ $?$ ” query, output a single line containing a single integer denoting the answer.

## Example

standard input	standard output
8 10	3
eldorado	4
? 1 3	6
? 1 8	6
# 6 7 it	6
? 1 8	6
# 3 3 t	
? 1 8	
# 1 8 streamer	
? 1 8	
# 1 8 symphony	
? 1 8	

## Note

- For the first query, the substring in consideration is “eld”. We must remove all three letters (turning it into the empty string) to make the string symphonic.
- For the second query, the substring in consideration is “eldorado”. We must remove at least 4 letters to make the string symphonic: eldorado → eldordo → eldodo → ldodo → dodo.
- After the third query, *s* becomes “eldorito”.
- For the fourth query, the substring in consideration is “eldorito”. We must remove at least 6 letters to make the string symphonic: eldorito → eldorto → eldoro → eldoo → edoo → edo → do.
- After the fifth query, *s* becomes “eltorito”.
- For the sixth query, the substring in consideration is “eltorito”. We must remove at least 6 letters to make the string symphonic: eltorito → eltorio → eltrio → eltri → etri → tri → ti.