

PaH TuNZ (remix)


Limite di tempo: 1.0 secondi
Limite di memoria: 256 MiB

Gabriele ha deciso di buttarsi nel mondo della musica trash. Per questo, ha scritto un software in grado di prendere un normale testo e remixarlo. Il processo è semplice, dal momento che tutto quello che fa il software è inserire nel testo un certo numero di effetti musicali, identificati dalle stringhe ‘PaH’ e ‘TuNZ’. Ad esempio, un testo formato dalle parole ‘Sette Otto’ può essere trasformato nel testo ‘TuNZTuNZSettePaHPaHTuNZOtto’. In generale, gli effetti sonori *possono* essere inseriti prima dell’inizio del testo e dopo la fine, ma non sono obbligatori, mentre tra due parole consecutive è sempre inserito almeno un effetto sonoro.

Giorgio non è un grande fan di questo genere musicale, e non riesce a distinguere bene le parole che formano il testo originale. Per questo è interessato a ripulire la musica dagli effetti sonori e ricostruire il testo originale. Aiutalo!

Implementazione

Dovrai sottoporre esattamente un file con estensione `.c`, `.cpp` o `.pas`.

 Tra gli allegati a questo task troverai un template (`remix.c`, `remix.cpp`, `remix.pas`) con un esempio di implementazione da completare.

Se sceglierai di utilizzare il template, dovrai implementare la seguente funzione:

C/C++	<code>void ripulisci(int N, char remix[], char testo[]);</code>
Pascal	<code>procedure ripulisci(N: longint; var remix, testo: array of char);</code>

In cui:

- L'intero N rappresenta il numero di caratteri del remix del testo.
- La stringa `remix`, indicizzata da 0 a $N - 1$, contiene il testo remixato.
- La funzione dovrà scrivere nella stringa `testo` il testo originale, che verrà stampato sul file di output.

Dati di input

Il file `input.txt` è composto da due righe. La prima riga contiene l'unico intero N . La seconda riga contiene la stringa `remix`.

Dati di output

Il file `output.txt` è composto da un'unica riga contenente una stringa, la risposta a questo problema.



Assunzioni

- $1 \leq N \leq 100\,000$.
- Il testo originale non contiene le parole ‘PaH’ e ‘TuNZ’.
- Il testo contiene almeno una parola.

Assegnazione del punteggio

Il tuo programma verrà testato su diversi test case raggruppati in subtask. Per ottenere il punteggio relativo ad un subtask, è necessario risolvere correttamente tutti i test relativi ad esso.

- **Subtask 1 [10 punti]:** Casi d’esempio.
- **Subtask 2 [20 punti]:** $N \leq 100$.
- **Subtask 3 [40 punti]:** $N \leq 1000$.
- **Subtask 4 [30 punti]:** Nessuna limitazione specifica.

Esempi di input/output

input.txt	output.txt
27 TuNZTuNZSettePaHPaHTuNZOtto	Sette Otto

input.txt	output.txt
12 PaHXXPaHTuNZ	XX