

Espressione di parentesi (parentesi)

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


Giorgio ha scritto un programma contenente una serie di espressioni molto elaborate, formate ciascuna da un gran numero di parentesi di tutti i tipi, e cioè:

- angolate: '<' e '>'
- tonde: '(' e ')'
- quadrate: '[' e ']'
- graffe: '{' e '}'

Purtroppo, quando ha provato ad eseguirlo, il compilatore gli ha detto che c'è un errore senza nemmeno dirgli in quale espressione si trova! Aiutalo controllando quali espressioni sono ben formate e quali no.

Implementazione

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

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

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

C/C++	<code>int controlla(int N, char E[]);</code>
Pascal	<code>function controlla(N: longint; var E: array of char): longint;</code>

In cui:

- L'intero N rappresenta la lunghezza dell'espressione da controllare.
- L'array E , indicizzato da 0 a $N - 1$, contiene i caratteri di cui l'espressione è composta.
- La funzione dovrà restituire 0 se l'espressione è corretta, -1 altrimenti. Nel primo caso, sul file di output verrà stampata la stringa `'corretta'`, nel secondo caso la stringa `'malformata'`.

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 E .

Dati di output

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



Assunzioni

- $1 \leq N \leq 10\,000$.
- E_i è uno tra i caratteri ' $\{[(\langle\rangle)]\}$ '.

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]:** Tutte le parentesi sono tonde.
- **Subtask 3 [20 punti]:** Le espressioni corrette sono formate da parentesi ordinate gerarchicamente (vedi quarto caso di esempio).
- **Subtask 4 [30 punti]:** $N \leq 30$.
- **Subtask 5 [20 punti]:** Nessuna limitazione specifica.

Esempi di input/output

input.txt	output.txt
4 ([)]	malformata
input.txt	output.txt
5 <({})	malformata
input.txt	output.txt
12 ()([]{(<>)})	corretta
input.txt	output.txt
20 {(<><>)}{({} [<>] <>) }	corretta

Spiegazione

Nel **primo caso di esempio**, la parentesi tonda più esterna viene chiusa prima della parentesi quadra (che è più interna), e quindi l'espressione è malformata.

Nel **secondo caso di esempio**, la parentesi angolata all'inizio non viene mai chiusa.

Nel **terzo e quarto caso di esempio**, tutte le parentesi sono chiuse correttamente. Inoltre nell'ultimo viene rispettata la gerarchia tra le parentesi: le angolate sono tutte più all'interno, a seguire tonde, quadre e infine graffe più all'esterno. In altre parole, una parentesi graffa non può essere aperta all'interno di un'altro tipo di parentesi; una parentesi quadra non può essere aperta all'interno di una parentesi tonda o angolata, e una parentesi tonda non può essere aperta all'interno di una parentesi angolata.