

Esercizio 3 (Parentesi Bilanciate II) Scrivere un programma che legga da `cin` una sequenza di caratteri fino ad incontrare la *sequenza* di caratteri `'e' 'n' 'd'`. Come per l'esercizio 2, dei caratteri letti deve considerare *solo* le parentesi tonde aperte `'('` e chiuse `)'` e deve verificare che le parentesi siano bilanciate, cioè che ogni parentesi aperta abbia una corrispondente parentesi chiusa e, viceversa, che ogni parentesi chiusa abbia una corrispondente parentesi aperta.

A fronte di una sequenza di parentesi bilanciate il programma deve scrivere: `"la sequenza e' bilanciata"`. Altrimenti, dovrà stampare: `"la sequenza non e' bilanciata: "`, seguita da **una** delle seguenti due frasi: `"ci sono parentesi aperte senza match"` oppure `"ci sono parentesi chiuse senza match"`, a seconda del caso.

Esempio 1: Supponiamo che i caratteri letti da `cin` siano i seguenti: `a b c) z e z z n (d) e n d`, allora la sequenza **non** è bilanciata. In questo caso, lo sbilanciamento viene scoperto alla prima parentesi chiusa letta. Quindi la lettura dei successivi caratteri non è necessaria e il programma dovrà scrivere in output: `"la sequenza non e' bilanciata: ci sono parentesi chiuse senza match"`. La seguente sequenza: `(a (e (n)) e n z (a)) e n d`, è invece bilanciata; in questo caso, il programma scriverà in output: `"la sequenza e' bilanciata"`. Un altro esempio di sequenza non bilanciata è il seguente `(a(b (c)) e d n (a) e n d`; il programma dovrà scrivere: `"la sequenza non e' bilanciata: ci sono parentesi aperte senza match"`. (Nota: negli 2 ultimi casi è necessario leggere tutti i caratteri fino alla sequenza di terminazione per sapere se la sequenza è bilanciata o meno).