

Esercizio 1. Si scriva un programma C++ che, letta da input una sequenza di caratteri (si intendono qui lettere minuscole dell'alfabeto) terminata dal carattere '.' ("punto"), stampi su standard output il numero di sottosequenze, definite come di seguito, che la compongono.

Ogni sottosequenza è composta da vocali e consonanti alternate, cioè una vocale seguita da una consonante, a sua volta seguita da una vocale, e così via. La sottosequenza può iniziare tanto con una vocale quanto con una consonante. Le sottosequenze sono separate le une dalle altre da coppie di vocali o coppie di consonanti: quindi, ogni volta che si incontrano due vocali di seguito (o, equivalentemente, due consonanti di seguito), termina una sottosequenza e ne inizia un'altra.

ESEMPIO: data la sequenza { **o** *g* *g* **i** *f* **a** *c* **a** / *d* **o** . } (in cui, per aiutare la comprensione, sono state evidenziate in grassetto le vocali ed in corsivo le consonanti), si individuano le sottosequenze: {og}, {gifacal}, {do} –Pertanto, il programma dovrebbe stampare su standard output il numero 3, pari al numero di sottosequenze presenti. Se invece la sequenza data in input è { **a** **a** **a** *b* *c* . }, allora si individuano le sottosequenze: {a}, {a}, {a b}, {c} e il programma dovrebbe stampare 4.

ATTENZIONE: Si noti che affinché il programma possa essere gestito dal valutatore automatico, deve essere inviato in stampa solo ed esclusivamente il numero di sottosequenze individuate (nel caso dell'esempio precedente, solo il numero 3), senza aggiungere endl, altre spaziature o altre stampe.