

Esercizio – conta ripetizioni parole (da “prova scritta del 2/9/2014”)

1) Sia `Elem` il tipo di una struttura dati (`struct`) costituita da due campi: `nome` di tipo stringa (lunghezza max. 100) e `cont` di tipo intero.

(a) Realizzare una funzione `void` di nome `aggiorna` che, presi come suoi parametri un array `A` di elementi di tipo `Elem`, il numero `m` di elementi in `A`, e una stringa `s`, controlla se `A` contiene già un elemento il cui campo `nome` coincide con `s`; in caso affermativo, incrementa di 1 il campo `cont` dell'elemento trovato; altrimenti aggiunge, dopo l'ultimo elemento inserito in `A`, un nuovo elemento con `nome` uguale a `s` e `cont` uguale a 1. La funzione restituisce in `m` il numero (eventualmente) aggiornato di elementi di `A`.

Ad es., dati `A={{“alfa”,2},{“beta”,1}}`, `m=2`, con `aggiorna(A,m,“beta”)` si ottiene `A={{“alfa”,2},{“beta”,2}}`, `m=2`, mentre con `aggiorna(A,m,“gamma”)` si ottiene `A={{“alfa”,2},{“beta”,2},{“gamma”,1}}`, `m=3`.

N.B. La funzione deve essere dichiarata obbligatoriamente di tipo `void`. Utilizzare soltanto stringhe “*tipo C*”.

(b) Scrivere un programma principale di prova che legge una sequenza di stringhe (max. 1000), separate tra loro da esattamente uno spazio, e crea un array `parole` di elementi di tipo `Elem`, utilizzando obbligatoriamente la funzione `aggiorna` applicata a ciascuna stringa letta. La lettura termina non appena si trova un “a capo” o si raggiunge il numero massimo di stringhe. Al termine della lettura il programma provvede a stampare l'array `parole` su std output, un elemento per ogni riga di stampa.

Prova esecuzione (input sottolineato)

Inserisci il testo, terminato da a capo (max. 100 parole):
il gatto e il cane e il topo

Numero di occorrenze di ciascuna parola nel testo:

```
il 3
gatto 1
e 2
cane 1
topo 1
```