

Prova scritta del 2/9/2014

*Convenzioni (obbligatorie). Utilizzare i nomi indicati nel testo -
 “Indentare” in modo opportuno i programmi - Inserire adeguati commenti*

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 `modifica` 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 in fondo ad `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`. N.B. La funzione deve essere dichiarata necessariamente di tipo `void`. Ad es., dati `A={{"alfa",2},{ "beta",1}}`, `m=2`, con `modifica(A,m,"beta")` si ottiene `A={{"alfa",2},{ "beta",2}}`, `m=2`, mentre con `modifica(A,m,"gamma")` si ottiene `A={{"alfa",2},{ "beta",2},{ "gamma",1}}`, `m=3`. N.B. Utilizzare soltanto stringhe “tipo C”.

(b) Descrivere la funzione `modifica` anche tramite diagramma di flusso.

2) Scrivere un programma principale che legge da un file, il cui nome è fornito dall'utente tramite `std input`, una sequenza di stringhe, separate da spazio, e crea un array `conta` di elementi di tipo `Elem` (max. 800), utilizzando obbligatoriamente la funzione `modifica` applicata a ciascuna stringa letta. Al termine della lettura dal file, il programma provvede a memorizzare l'array `conta` su un file il cui nome è ottenuto dal file di input preceduto dalla stringa `"ris_per_"` e quindi a stamparlo su `std output`. N.B. Il nome del file di input può avere lunghezza massima di 50 caratteri e può contenere “spazi”. Controllare l'apertura del file e la situazione in cui l'array `conta` risulti “pieno”. Es.:

File di input `prova1.txt`:

`il gatto e il cane e il topo`

File di output `ris_per_prova1.txt`:

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

3) Realizzare una funzione `void` di nome `inverti` che, preso come suo (unico) parametro una lista concatenata semplice di numeri interi `l`, modifica `l` in modo tale che l'ordine dei suoi elementi risulti inverso rispetto a quello originale. Ad es., `l={1,2,3}` diventa `l={3,2,1}`. SUGG. Utilizzare una lista temporanea `aux` in cui effettuare l'inserimento in testa gli elementi via via estratti da `l` ...