

Prova scritta del 9/1/2006

1) Sia `Elem` il tipo di una struttura (`struct`) costituita da due campi, `C1` di tipo reale, e `C2` di tipo stringa (array di caratteri), di lunghezza massima 32.

(a) Realizzare una funzione di nome `TROVA` che, presi come suoi parametri un array `Tab` di strutture di tipo `Elem`, il numero `nTab` di elementi in `Tab`, ed una stringa `Nome`, verifica se `Tab` contiene almeno un elemento con campo `C2` uguale a `Nome`. La funzione restituisce l'indice dell'elemento trovato, oppure `-1` se non trova alcun elemento con la proprietà richiesta.

(b) Realizzare una funzione di nome `SOMMA` che, presi come suoi parametri due array `A` e `B` di strutture di tipo `Elem`, ed il numero `nA` di elementi in `A`, memorizza in `B`, per ogni diverso valore `s` del campo `C2` degli elementi di `A`, un elemento `{T, s}` in cui `T` è ottenuto come somma di tutti i valori dei campi `C1` degli elementi di `A` aventi lo stesso `s`. Ad esempio, se `A` contiene `{{1.1, "alfa"}, {2.5, "beta"}, {2.1, "beta"}, {1.8, "alfa"}}`, `B` dovrà contenere `{{2.9, "alfa"}, {4.6, "beta"}}`. La funzione restituisce il numero di elementi memorizzati in `B`. Si richiede di utilizzare obbligatoriamente la funzione `TROVA` all'interno della funzione `SOMMA` (SUGG.: usare la `TROVA` per verificare se un certo elemento di `A` è già stato inserito o no in `B`, ...).

(c) Descrivere la funzione `TROVA` anche tramite un diagramma di flusso.

2) Scrivere un programma principale che, preso un file di nome "entrate", contenente una sequenza di coppie costituite ciascuna da un importo in euro (un numero reale) ed un nome di persona (una stringa di max 32 caratteri), legge la sequenze dal file in un array di strutture di tipo `Elem` di nome `E`, memorizzando ciascuna coppia della sequenza in un elemento dell'array `E`. Il programma quindi calcola un nuovo array `ETot`, contenente le somme degli importi per ciascuna persona in `E`, applicando (obbligatoriamente) la funzione `SOMMA` all'array `E`. Infine memorizza l'array `ETot` su un nuovo file di nome "entrate_tot" con lo stesso formato del file "entrate".

3) Modificare la definizione della funzione `SOMMA` in modo tale che il suo tipo sia `void` e che il numero degli elementi memorizzati in `B` dalla funzione sia un parametro della funzione stessa passato per riferimento. Mostrare anche le parti del codice del programma principale che devono essere modificate per utilizzare la nuova versione della funzione `SOMMA`.