

Prova scritta del 18/9/2013

Strutturare adeguatamente i programmi ed evidenziarne la strutturazione mediante indentazione. Inserire anche adeguati commenti

1) Sia `Dati` il tipo di una struttura dati `struct` costituita da due campi `d` ed `s`, rispettivamente di tipo intero e stringa (di lunghezza massima 80).

(a) Realizzare una funzione di nome `trova` che, presi come suoi parametri un array `A` di strutture di tipo `Dati`, il numero `n` di elementi in `A`, una stringa `r`, ed un array di interi `Ris`, memorizza in `Ris` l'indice di tutti gli elementi di `A` in cui la stringa contenuta nel campo `s` precede `r` secondo il consueto ordine lessicografico. La funzione restituisce come suo risultato il numero di elementi memorizzati in `Ris`. N.B. Utilizzare soltanto stringhe "tipo C".

(b) Realizzare un programma principale di prova che richiami la funzione `trova` passandole i valori `{{27, "ddd"}, {30, "aaa"}, {40, "fff"}, {35, "bbb"}}`, `4`, `"ccc"`, rispettivamente in corrispondenza ai parametri `A`, `n`, ed `r`. Il programma quindi stampa su `std output` il valore dei campi `d` degli elementi individuati (risultato atteso: `30 35`).

2) Scrivere un programma principale che: (a) dichiara una matrice `M` di elementi di tipo numero reale di dimensioni `10 x 10` e li inizializza tutti con il valore `0`; (b) legge da un file di nome `"dati.txt"` una sequenza di numeri reali `e`, per ciascun numero letto, chiede all'utente di fornire tramite tastiera due numeri `i` e `j` e quindi memorizza il numero letto in `Mij`; (c) al termine della lettura dei numeri dal file stampa su `std output` l'intera matrice `M`. N.B. Se il file non esiste, il programma termina immediatamente. Se i valori degli indici forniti dall'utente non sono corretti, oppure se l'elemento di `M` individuato contiene già un numero diverso da `0`, il programma richiede all'utente di ripetere l'immissione dei dati.

3) Sia `elem` il tipo così definito: `struct elem {int info; elem* punt;}` e sia `Elenco` un array di `n` liste concatenate semplici i cui elementi sono di tipo `elem` ($n \leq 10$). Scrivere una funzione booleana di nome `estrai_testa` che, presi come suoi parametri l'array `Elenco` e il numero `n`, estrae l'elemento di testa della prima lista non vuota che incontra in `Elenco`. Se nessuna lista in `Elenco` contiene elementi, la funzione restituisce `false`; altrimenti restituisce `true`. In quest'ultimo caso la funzione restituisce anche, come suo ulteriore parametro, il valore del campo `info` dell'elemento estratto. N.B. La funzione provvede anche a deallocare la memoria dell'eventuale elemento estratto