

Prova scritta del 23/2/2010

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

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

- 1) (a) Realizzare una funzione di nome `trova_tutti` che, presi come suoi parametri un array `A` di strutture di tipo `Dati`, il numero `n` di elementi in `A`, un numero intero `m`, ed un array di interi `Ris`, memorizza in `Ris` gli indici di tutti gli elementi di `A` il cui valore del campo `d` è non nullo e $\leq m$. La funzione restituisce come suo risultato il numero di elementi che sono stati memorizzati in `Ris`. Ad es., se `A` è `[(30.5, "aaa"), (40.1, "bbb"), (35.0, "ccc"), (0, "ddd")]` e `m = 35`, l'array `Ris` sarà `[0, 2]`. N.B. Usare soltanto stringhe del C.
 (b) Descrivere la funzione anche tramite diagramma di flusso.

2) Scrivere un programma principale che: (a) dichiara una matrice `M` di elementi di tipo `Dati` di dimensioni `10 x 10` ed inizializza a 0 i campi `d` di tutti gli elementi di `M`; (b) legge da un file di nome `"mappa.txt"` una sequenza di quadruple `<i,j,k,f>`, dove `i,j` sono numeri interi, `k` è un numero reale e `f` è una stringa, e, per ciascuna quadrupla letta, memorizza `k` ed `f` rispettivamente nei campi `d` ed `s` dell'elemento `M(i,j)`; (c) chiede all'utente di fornire tramite tastiera un numero intero `r` ($0 \leq r < 10$) e un numero intero `lim` (> 0) e quindi, utilizzando (obbligatoriamente) la funzione `trova_tutti`, determina tutti gli elementi della riga `M(r)` con valore del campo `d` $\leq \text{lim}$ e li memorizza in un array `t`; (d) stampa su std output l'array `t` (solo gli elementi "significativi" di `t`!). N.B. Si assuma che il file esista, che i dati in esso contenuti siano corretti e che ciascuna quadrupla sia separata dalla successiva da un "a capo". Si controlli invece la correttezza dei dati forniti dall'utente (punto (c)) e in caso di dato non corretto si ripeta l'operazione di lettura.

3) Scrivere un programma principale che confronta due file di caratteri e determina se i due file sono identici oppure no. In quest'ultimo caso, il programma stampa il numero della riga e la riga stessa in cui viene riscontrata la prima differenza tra i due file (per riga si intende una sequenza qualsiasi di caratteri – anche vuota – terminata dal carattere speciale `'\n'`). Altrimenti, stampa il messaggio che i due file sono identici. Il nome del primo file è fornito dall'utente tramite tastiera, mentre il nome del secondo file è ottenuto da quello del primo aggiungendovi il prefisso `"nuovo_"` (ad es., se il nome del primo file è `"prova.txt"`, il nome del secondo sarà `"nuovo_prova.txt"`). Controllare anche l'esistenza dei due file e in caso negativo terminare il programma.