

Prova scritta del 27/2/2007

Strutturare adeguatamente i programmi ed evidenziarne la strutturazione mediante indentazione. Inserire anche adeguati commenti su istruzioni e dati utilizzati e sulle scelte algoritmiche effettuate

1) Sia `S` un tipo `struct` composto da tre campi di tipo `float`, di nome `Entrate`, `Uscite`, `Saldo`, ed un campo di nome `data_giorno` di tipo `data`, dove `data` è un tipo `struct` di tre campi interi di nome `g`, `m`, `a`. Scrivere un programma che esegua nell'ordine le seguenti azioni: (1) chiede all'utente l'anno ed il numero d'ordine del mese corrente, controllando l'ammissibilità di quest'ultimo; (2) chiede all'utente il nome del file di input e lo memorizza nella variabile `nomefile`; (3) memorizza in un array di strutture di tipo `S`, di nome `movimenti` (max. 31 elementi), i dati prelevati dal file `nomefile` utilizzando la funzione `carica_file` (vedi sotto); se la funzione restituisce `false` il programma termina immediatamente; (4) calcola la somma totale dei campi `saldo` di tutti gli elementi (significativi) dell'array `movimenti` e stampa il risultato; (5) apre un file di output il cui nome è ottenuto tramite la funzione `componi_nome` (vedi sotto), passandole come primo parametro il nome del file di input e come secondo parametro la stringa "guadagni" o "perdite" a seconda che il totale calcolato al punto (4) sia o no positivo (il terzo parametro di `componi_nome` conterrà il nome costruito dalla funzione); (6) scrive sul file di output i campi `saldo` e `g` del campo `data_giorno` di tutti gli elementi (significativi) di `movimenti` (inserire un "a capo" dopo ogni elemento di `movimenti` scritto sul file); (7) termina.

2) Realizzare una funzione booleana di nome `carica_file` che, presi come suoi parametri una stringa `N`, un array `A` di strutture di tipo `S`, e due interi `M` ed `Y`, legge dal file di nome `N` una sequenza di numeri reali e li memorizza, due a due, nei campi `entrate` e `uscite` di ciascun elemento di `A`. La funzione inoltre, per ciascun elemento di `A`, memorizza nel campo `saldo` la differenza tra `entrate` ed `uscite` e nei campi `g`, `m` ed `a` del campo `data_giorno`, rispettivamente, l'indice dell'elemento stesso incrementato di 1 e gli interi `M` ed `Y`. La funzione restituisce come suo risultato `false` se l'operazione di apertura del file `N` non ha avuto successo; `true` altrimenti. In quest'ultimo caso la funzione restituisce anche, come suo ulteriore parametro di nome `n`, il numero di elementi inseriti in `A` (n.b., se il numero di elementi presenti sul file supera 31, gli elementi in eccedenza vengono ignorati).

3) Scrivere una funzione `void`, di nome `componi_nome` che, presi come suoi parametri tre stringhe `S1`, `S2` ed `S3`, memorizza in `S3` la stringa ottenuta concatenando `S1` ad `S2` ed inserendo tra le due stringhe il carattere '_'. Ad esempio, se `S1 = "dati.txt"` e `S2 = "perdite"`, la stringa `S3` sarà: "perdite_dati.txt". Si richiede (obbligatoriamente) di non utilizzare le funzioni della libreria `cstring`.

N.B. Utilizzare esclusivamente stringhe del C, ovvero array di caratteri terminati da '`\0`'.