

Prova scritta del 21/9/2012

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

1) (a) Si supponga di rappresentare un intervallo chiuso di interi come un tipo `struct` di nome `Intv` costituito da due campi interi, `inf` e `sup`, contenenti rispettivamente il limite inferiore e quello superiore dell'intervallo. Realizzare una funzione booleana di nome `interseca` che, presi come suoi parametri un array `I` di intervalli e la sua dimensione `n`, restituisce `true` se l'intersezione degli intervalli in `I` è non vuota, `false` altrimenti. Nel primo caso, la funzione restituisce anche, come suo ulteriore parametro di nome `Ris`, l'intervallo ottenuto dall'intersezione degli intervalli in `I`. Ad esempio, se `I` è $\{\{3,7\}, \{2,6\}, \{5,10\}\}$, `Ris` sarà $\{5,6\}$.

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

2) Scrivere un programma principale che: (1) presenta all'utente (su standard output) un menù di possibili operazioni su intervalli (realizzati come nell'esercizio (1)); (2) esegue l'operazione scelta; (3) ripete da (1) finché non viene scelta l'operazione di "smetti". Le possibili operazioni sono:

a. richiede all'utente un intervallo (estremo inferiore e superiore) e lo memorizza in un array di intervalli di nome `Dati` (di dim. max. 1000), di seguito agli eventuali altri intervalli già inseriti; se si supera la capacità massima dell'array, l'intervallo dato non viene inserito e viene stampato un opportuno messaggio;

b. determina e stampa l'intersezione di tutti gli intervalli memorizzati in `Dati` oppure un opportuno messaggio se l'intersezione è vuota (si utilizzi obbligatoriamente la funzione `interseca` dell'esercizio (1));

c. smetti.

3) Scrivere un programma principale che legge da due file distinti due sequenze di numeri reali e per ogni coppia di numeri corrispondenti (= stessa posizione nei due file) scrive su un terzo file la loro somma. La lettura termina appena si raggiunge l'*end-of-file* di uno dei due file di input. Ad esempio, se il primo file contiene [2.1 4.5 3.0] e il secondo contiene [5.3 4.0], il file di output conterrà [7.4 8.5]. I nomi dei file di input sono forniti all'inizio del programma dall'utente tramite standard input (massimo 32 caratteri); il nome del file di output è invece ottenuto concatenando i nomi dei file di input e la stringa "somma" nel modo seguente: se "alfa" e "beta" sono i nomi dei file di input il nome del file di output sarà "somma_alfa_beta". SUGG.: non memorizzare i numeri letti dai file di input in un array, ma eseguire direttamente la somma per ogni coppia di numeri letti.