

Prova scritta del 9/7/2009

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

1) Sia `interv` il tipo di una struttura dati `struct` costituita da due campi interi, `inf` e `sup`, che rappresentano rispettivamente il limite inferiore e quello superiore di un intervallo chiuso $[inf, sup]$. Realizzare una funzione booleana di nome `aggiungi_elemento` che, presi come suoi parametri una struttura `I` di tipo `interv` e un intero `a`, aggiunge, se possibile, l'elemento `a` all'intervallo `I`, modificando (eventualmente) `I` stesso. L'operazione è possibile o quando `a` è contenuto nell'intervallo `I` (`I` resta invariato) o quando `a` coincide con il predecessore di `inf` o con il successore di `sup` di `I` (`I` viene modificato aggiornando rispettivamente `inf` o `sup`). In questi casi la funzione restituisce `true`; altrimenti restituisce `false`. Ad es., se `I = [3, 7]` e `a = 8`, il nuovo valore di `I` sarà `[3, 8]`, mentre se `a = 9`, l'operazione non è possibile (risultato `false`). N.B. Attenzione alla modalità di passaggio del parametro `I` ...!

2) (a) Realizzare una funzione di nome `aggiorna_intervalli` che, presi come suoi parametri un array `U` di strutture di tipo `interv`, il numero `n` di elementi in `U`, e un intero `x`, cerca in `U` (se esiste) un intervallo `I` in cui sia possibile inserire `x` e, nel caso lo trovi, vi inserisce `x` come specificato nell'esercizio 1 (si richiede di utilizzare obbligatoriamente la funzione `aggiungi_elemento`). Altrimenti, aggiunge l'intervallo contenente solo `x` (ovvero $[x, x]$) come ultimo elemento di `U`. Ad es., se `U = ([1, 6], [10, 15])` e `x = 9`, il nuovo `U` sarà `([1, 6], [9, 15])`, mentre se `x = 8`, il nuovo `U` sarà `([1, 6], [10, 15], [8, 8])`.

(b) Realizzare anche un programma principale di prova che dichiara un array `A` di `interv` di dimensione massima 10, lo inizializza con la sequenza di intervalli `([1, 6], [10, 15])`, vi aggiunge l'elemento 9 tramite la funzione `aggiorna_intervalli`, e quindi stampa la nuova sequenza di intervalli contenuta in `A`.

3) Scrivere un programma principale che richiede all'utente il nome di due file (lunghezza massima del nome, 64 caratteri), legge ordinatamente i dati del primo e del secondo file e li memorizza (nello stesso ordine) in un nuovo file di output. Il nome del file di output è ottenuto concatenando i nomi dei due file di input, separati da un carattere `'_'`. Ad esempio, se i nomi dei file di input sono `"file1.txt"` e `"file2.txt"` il nome del file di output sarà `"file1.txt_file2.txt"`. N.B. Non si fanno assunzioni sulle dimensioni massime dei file. Si richiede di utilizzare soltanto stringhe tipo C (= array di caratteri terminati da `'\0'`)..