Algoritmica – Prova di Laboratorio

Corso A e B

Appello del 08/07/2011

Istruzioni

Risolvete il seguente esercizio, prestando particolare attenzione alla formattazione dell'input e dell'output, in quanto la correzione avverrà in maniera automatica.
Per consegnare un elaborato dovete fornire il codice sorgente attraverso il comando `./consegna` che avete nella vostra home directory e che provvede ad inviare il vostro esercizio al server di valutazione:

```
./consegna sorgente.c 1
```

dove `sorgente.c` è il nome del file che contiene la soluzione che avete elaborato, ricordando che il percorso deve essere specificato a partire dalla vostra home directory. Il numero che segue identifica il numero dell'esercizio, che in questa prova è solo il numero `1`.
Il comando `consegna` può essere utilizzato molteplici volte, per cui è possibile sovrascrivere la propria soluzione per un dato esercizio. Di tutte le consegne per un dato esercizio, viene corretta soltanto l'ultima. Il file da consegnare deve contenere nelle prime righe un commento `C` che specifica il vostro Nome, Cognome e Numero di Matricola. Per esempio:

```
/*
    Nome: Alan
    Cognome: Turing
    Matricola: 193700
*/
```

Lo script di consegna, prima di inviare la vostra soluzione al server, proverà ad eseguire il vostro codice utilizzando alcuni file di input predefiniti e controllerà la correttezza dell'output prodotto. Tali file di input e output per la verifica del codice sono stati collocati nella cartella `dati/1/` della vostra home directory. I file di input e output per effettuare il test dell'esercizio sono nominati secondo lo schema: `input0.txt output0.txt input1.txt output1.txt input2.txt output2.txt ...` e possono essere usati anche da voi per testare il vostro codice e verificare poi manualmente la correttezza dell'output prodotto. Ad esempio:

```
./compilato < dati/1/input0.txt
```

effettua il test del vostro codice sui dati contenuti nel primo file di input, assumendo che `compilato` contenga la compilazione della vostra soluzione e che si trovi nella vostra home directory. Dovete aspettarvi che l'output corrisponda a quanto contenuto nel file `dati/1/output0.txt`. Per effettuare un controllo in automatico sul primo file input `input0.txt` potete eseguire questi due comandi in sequenza:

```
./compilato < dati/1/input0.txt > res0
diff res0 dati/1/ouput0.txt
```

Il primo comando infatti esegue la vostra soluzione e stampa l'output prodotto nel file `res0`, il secondo infine controlla le differenze fra l'output prodotto da voi e quello corretto.

La consegna andrà a buon fine solo se il vostro codice riesce a superare tutti i test contenuti nella cartella `dati`. Eventualmente lo script di consegna vi informa per quali di questi test il vostro codice non risponde correttamente. Nel caso invece la vostra soluzione passi i test, vi verrà chiesto di specificare il vostro numero di matricola e la soluzione sarà consegnata. Una volta effettuata la prima consegna, il numero di matricola verrà associato alla macchina e tramite quella macchina non sarà possibile consegnare altri elaborati se non per quello specifico numero di matricola.
Una volta consegnata, la vostra soluzione verrà valutata nel server di consegna utilizzando altri file di test, da voi non accessibili. Il consiglio è quindi quello di non provare ciecamente e ripetutamente la vostra soluzione con i file di test che vi sono forniti ma cercare di ragionare sul codice che avete scritto perché il fatto che il vostro codice passi i test di consegna non significa necessariamente che questo sia corretto in assoluto.

Consiglio: per controllare il proprio codice si consiglia di inserire delle chiamate di *test* alla `printf` per permettervi di visualizzare lo stato del programma, il valore di alcune variabili, ecc. Ognuna di queste chiamate deve essere immediatamente seguita dal comando `fflush(stdout);` per garantire che venga visualizzato il prodotto della `printf`. Una volta completato l'esercizio, tutte le chiamate di *test* devono essere rimosse affinchè l'output corrisponda strettamente a quanto richiesto.

Esercizio 1

È data in input una sequenza di N interi positivi dalla quale deve essere costruito un albero binario di ricerca inserendo i nodi nell'ordine in cui sono forniti e senza effettuare operazioni di bilanciamento dell'albero. Un nodo dell'albero è detto PD se soddisfa le seguenti proprietà :

- il nodo ha due figli;
- tutti i nodi appartenenti al suo sottoalbero *sinistro* contengono chiavi *pari*;
- tutti i nodi appartenenti al suo sottoalbero *destro* contengono chiavi *dispari*.

Scrivere quindi un programma che legga la sequenza di input, costruisca l'albero binario di ricerca secondo le specifiche e stampi la chiave del nodo PD di valore massimo all'interno dell'albero, o il valore `-1` se non esistono nodi PD nell'albero.
L'input è formattato nel seguente modo. Nella prima riga è contenuto l'intero N. Seguono poi N righe contenenti ognuna un intero appartenente alla sequenza con cui costruire l'albero. Si può assumere che la sequenza contenga interi positivi strettamente maggiori di zero e tutti distinti.
L'output invece è costituito da una singola riga che contiene il risultato del programma, seguito da un carattere di ritorno a capo `\n`.

Esempio

Notare che nell'esempio qui riportato, gli unici nodi PD sono quelli con chiave 11 e 20 e di conseguenza l'output corretto è `20\n`.

Input	**Output**	**Albero**

`8`
`16`
`11`
`20`
`18`
`15`
`4`
`2`
`39`
	`20`