

# Algoritmi e Strutture Dati – Prova di Laboratorio

11/01/2016

## Istruzioni

Risolvere il seguente esercizio prestando particolare attenzione alla formattazione dell'input e dell'output. La lettura dell'input e la scrittura dell'output **DEVONO** essere effettuate tramite gli stream **cin** e **cout** rispettivamente. La correzione avverrà in maniera automatica eseguendo dei test e confrontando l'output prodotto dalla vostra soluzione con l'output atteso. Si ricorda che è possibile verificare la correttezza del vostro programma su un sottoinsieme dei input/output utilizzati. I file di input e output per i test sono nominati secondo lo schema: `input0.txt output0.txt input1.txt output1.txt ...`. Per effettuare le vostre prove potete utilizzare il comando del terminale per la redirectione dell'input. Ad esempio

```
./compilato < 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 coincida con quello contenuto nel file `output0.txt`. Per effettuare un controllo automatico sul primo file input `input0.txt` potete eseguire la sequenza di comandi

```
./compilato < input0.txt | diff - output0.txt
```

Questa esegue la vostra soluzione e controlla le differenze fra l'output prodotto e quello corretto.

Una volta consegnata, la vostra soluzione verrà valutata nel server di consegna utilizzando altri file di test non accessibili. Si ricorda di avvisare i docenti una volta che il server ha accettato una soluzione come corretta.

## Esercizio

Scrivere un programma che legga da tastiera una sequenza di  $N$  coppie di interi  $[valore, ID]$  e le utilizzi per riempire  $D$  alberi binari di ricerca (**senza** ribilanciamento). Ciascun albero è caratterizzato da un  $ID$  intero compreso tra 0 e  $D - 1$ . Per ciascuna coppia letta, il primo intero indica il valore che deve essere inserito, mentre il secondo intero indica l' $ID$  dell'albero di destinazione. I valori devono essere inseriti nello stesso ordine con cui vengono letti. (per convenzione, i valori  $\leq$  vanno inseriti a sinistra)

Il programma deve:

- identificare *low* come l' $ID$  dell'albero con la più piccola altezza. A parità di altezza, scegliere l'albero con ID minore.
- identificare *high* come l' $ID$  dell'albero con la più grande altezza. A parità di altezza, scegliere l'albero con ID maggiore.
- considerare i nodi dell'albero con ID *low* in ordine crescente e inserirli nell'albero con ID *high*
- stampare in ordine crescente le etichette delle foglie dell'albero risultante

**NOTA:** L'altezza del nodo radice è 0.

L'**input** è formattato nel seguente modo: la prima riga contiene gli interi  $N$  e  $D$  separati da uno spazio. Seguono  $N$  righe contenenti una coppia  $[intero, intero]$  ciascuna, con gli elementi della coppia separati da uno spazio.

L'**output** contiene gli elementi della soluzione, uno per riga.

## Esempio

### Input

```
6 2
6 0
2 0
5 1
4 1
10 0
3 0
```

### Output

```
5
10
```

