

# Algoritmica – Esame di Laboratorio

14/06/2013

## Istruzioni

Risolvete il seguente esercizio prestando particolare attenzione alla formattazione dell'input e dell'output. 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 corrisponda a quanto contenuto nel file `output0.txt`. Per effettuare un controllo automatico sul primo file input `input0.txt` potete eseguire i comandi:

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

Il comando 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 1

Scrivere un programma che legga da tastiera un intero  $N$  e una sequenza di  $N$  coppie *chiave e valore*. Le  $N$  chiavi sono interi positivi, **NON** necessariamente distinti. I valori sono stringhe di lunghezza al più 100 caratteri. Il programma deve inserire uno alla volta, nell'ordine dato, le coppie in una tabella hash  $T$  di dimensione  $2N$ . La risoluzione di eventuali conflitti avviene con liste monodirezionali. Per ciascuna coppia, l'elemento corrispondente nella lista memorizza sia la chiave che il valore. Per l'inserimento si deve utilizzare la funzione:

- $h(x) = (x \% 109) \% 2N$

dove  $x$  è la chiave di una coppia.

Nell'inserimento di una coppia con chiave  $x$  si deve **sempre** controllare che non sia già presente una coppia avente la stessa chiave  $x$ . In questo caso infatti la vecchia coppia deve essere **sostituita** con la nuova.

Al termine dell'inserimento, il programma deve leggere un intero  $K$  e stampare i valori delle coppie nella  $K$ -esima lista, **ordinate** lessicograficamente.

L'input è formattato nel seguente modo. La prima riga contiene l'intero  $N$ . Seguono poi  $2N$  righe, due righe per coppia. La prima riga della coppia contiene la chiave, mentre la seconda contiene il valore. L'ultima riga dell'input contiene il valore  $K$ .

L'output deve riportare i valori delle coppie presenti nella  $K$ -esima lista della tabella  $T$ , ordinati lessicograficamente e stampati uno per riga. Il programma stampa *vuota* se la  $K$ -esima lista è vuota.

## **Esempio**

### **Input**

10  
109  
TyrionLannister  
3  
NedStark  
4  
RobertBaratheon  
3  
KhalDrogo  
0  
CatelynStark  
0  
KingJoffrey  
110  
JonSnow  
0  
DaenerysTargaryen  
1  
WinterIsComing  
6  
Hodor  
0

### **Output**

DaenerysTargaryen  
TyrionLannister