

Algoritmi e Strutture Dati – Prova di Laboratorio

27/01/2016

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

Si consideri un sistema bancario che memorizza i dati relativi a conti correnti. Ciascun conto corrente è identificato da un **ID** univoco e intero, e da una stringa **Cognome** che ne specifica il titolare. Il sistema mantiene questi dati dentro una tabella hash con liste di trabocco (metodo di concatenazione). Scrivere un programma che

- legga da tastiera una sequenza di N coppie (**ID**,**Cognome**) ciascuna rappresentante un conto corrente;
- salvi dentro una *tabella Hash* le informazioni relative ai conti correnti, utilizzando la seguente funzione hash:

$$h(ID) = \{ [(a \times ID) + b] \% p \} \% 2N$$

dove $p=999149$, $a=1000$ e $b=2000$;

- identifichi l'indirizzo della *tabella Hash* che ha generato più collisioni. A parità di numero di collisioni scegliere l'indirizzo con indice minore;
- dato l'indirizzo ottenuto al passo precedente, stampi l'**ID** del primo elemento secondo l'ordine lessicografico per **Cognome**. A parità di **Cognome**, scegliere l'elemento con **ID** più basso.

L'**input** è formattato nel seguente modo: la prima riga contiene l'intero N . Seguono N righe contenenti una coppia (**intero**,**stringa**).

L'**output** contiene la soluzione.

Esempio

Input

```
5
43047 Kilmister
43046 Bowie
58 Frey
1957 Scola
1900 Rickman
```

Output

```
1900
```