

Algoritmica – Esame di Laboratorio

27/06/2014

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 degli 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 di 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 vuole implementare una rubrica telefonica, in cui un record della rubrica è costituito da un nome (una stringa senza spazi, univoca) a cui è associato un numero di telefono. Sia N il numero di record da inserire nella rubrica. Essi devono essere memorizzati in una tabella hash di dimensione $2N$.

La funzione hash riceve in input una stringa non vuota (il nome) e restituisce in output un valore numerico:

- si esegue la somma di tutti i codici decimali associati ai caratteri della stringa (escluso il carattere speciale di fine stringa '\0'); ogni carattere **deve** essere trattato come `unsigned int`;
- l'output è pari alla somma precedentemente calcolata modulo $2N$.

Ad esempio, per la stringa `Ciao!` ($N = 5$) si hanno i codici indicati nella seguente tabella:

Carattere	Codice decimale
C	67
i	105
a	97
o	111
!	33

Si ottiene: $67 + 105 + 97 + 111 + 33 = 413$, $2N = 10$ e dunque $\text{hash}(\text{Ciao!}) = 413 \% 10 = 3$.

Quando si inserisce un record in una lista di trabocco, se vi sono conflitti il record va inserito in ordine lessicografico non decrescente, scandendo opportunamente la lista fino ad individuare la posizione in cui inserirlo.

Dato un intero K tale che $0 \leq K < 2N$, il programma deve stampare il contenuto della lista di trabocco di posizione K . Data la maniera in cui vengono inseriti i record nelle liste, l'output sarà costituito da record stampati in base all'ordine lessicografico del nome.

NOTA IMPORTANTE: Non è consentito l'utilizzo di alcun algoritmo di ordinamento al fine di riordinare i record della K -esima lista. La procedura di inserimento di un record è sufficiente a garantire l'ordinamento.

L'input è così formato:

- la prima riga contiene il numero N di record da inserire;
- le successive $2N$ righe contengono la descrizione degli N record; la descrizione di un record occupa 2 righe consecutive:
 - la prima contiene il nome (una stringa non vuota, univoca, di lunghezza non superiore a 100 caratteri);

– la seconda contiene il numero di telefono (una stringa non vuota di lunghezza non superiore a 20 caratteri);

- l'ultima riga contiene il valore K .

L'output è costituito dalla sequenza dei record presenti nella K -esima lista di trabocco, stampati in ordine lessicografico del nome, uno per riga. Ogni record è stampato nel formato:

<nome> <spazio> <numero di telefono>

(ad esempio AlanTuring 123-456789).

Esempio

Input

```
4 // N
Gianni // pos. 6
050-123456
Carlo // pos. 1
050-234567
Maria // pos. 2
050-345678
Antonella // pos. 6
050-456789
6 // K
```

Output

```
Antonella 050-456789
Gianni 050-123456
```