

Algoritmica – Esame di Laboratorio

13/06/2014

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 aiutare uno studente ad organizzare il suo piano di studi universitario. Il corso di laurea non prevede esami obbligatori, tutti gli esami sono opzionali. Lo studente soffre di eccessivo stress da esame, quindi l'obiettivo prioritario è minimizzare il numero di esami. Esiste una lista di N esami sostenibili. Lo studente deve scegliere il numero di esami in maniera da arrivare ad un numero di crediti formativi *uguale* a K . Ogni esame è caratterizzato da:

- sigla dell'esame (una stringa non vuota di lunghezza non superiore a 100 caratteri);
- numero di crediti (un intero positivo);
- grado di difficoltà (un intero positivo; più l'intero è grande, più l'esame è difficile).

Lo studente vuole minimizzare il numero di esami. A parità di crediti per esame, vuole minimizzare la difficoltà. Scrivere un programma che opera come segue:

- si selezionano inizialmente gli esami con il maggior numero di crediti; a parità di crediti, si selezionano prima gli esami aventi difficoltà minore; a parità di difficoltà, si selezionano gli esami in base all'ordine lessicografico non decrescente della sigla;
- se l'inserimento di un esame provoca il superamento dei K crediti, viene scartato.

NOTA IMPORTANTE: tutte le istanze di test su cui l'algoritmo verrà valutato sono costruite in maniera da raggiungere **esattamente** i K crediti mediante la procedura appena descritta.

NOTA: La procedura descritta è un'euristica per il problema dello zaino.

Il programma deve stampare la lista degli esami scelti in ordine lessicografico.

L'input è così formato:

- la prima riga contiene il numero K di crediti da raggiungere ($K > 0$);
- la seconda riga contiene il numero N di esami possibili tra cui scegliere ($N > 0$);
- le successive $3N$ righe contengono la descrizione degli N esami; la descrizione di un esame occupa 3 righe consecutive:
 - la prima contiene la sigla dell'esame (una stringa non vuota di lunghezza non superiore a 100 caratteri;

- la seconda contiene il numero di crediti dell'esame (un intero positivo);
- la terza contiene il grado di difficoltà dell'esame (un intero positivo; più l'intero è grande, più l'esame è difficile).

L'output è costituito dalla sequenza delle sigle degli esami selezionati, stampate in ordine lessicografico, una per riga.

Esempio

Input

30 // K
6 // N
AB
12
15
CD
6
10
EF
12
8
GH
12
9
LM
6
9
NP
9
14

Output

EF
GH
LM

Si noti che, ordinando in base al criterio stabilito (numero di crediti, difficoltà, sigla), la lista degli esami risulta: EF, GH, AB, NP, LM, CD.