

# Algoritmica – Prova di Laboratorio

Corso A e B

Appello del 12/09/2011

## Istruzioni

Risolvete il seguente esercizio, prestando particolare attenzione alla formattazione dell'input e dell'output, in quanto la correzione avverrà in maniera automatica.

Per consegnare un elaborato dovete fornire il codice sorgente attraverso il comando `./consegna` che avete nella vostra home directory e che provvede ad inviare il vostro esercizio al server di valutazione. Il comando deve essere utilizzato dal vostro terminale nel seguente modo:

```
./consegna sorgente.c numEx
```

dove:

- `sorgente.c` è il nome del file che contiene la soluzione che avete elaborato, ricordando che il percorso deve essere specificato a partire dalla vostra home directory
- `numEx` è l'identificativo numerico dell'esercizio a cui fa riferimento la soluzione (in questa prova l'esercizio è solo il numero 1, per cui al posto di `numEx` dovete mettere sempre 1).

Il comando `consegna` può essere utilizzato molteplici volte, per cui è possibile sovrascrivere la propria soluzione per un dato esercizio. Di tutte le consegne per un dato esercizio, viene corretta soltanto l'ultima. Il file da consegnare deve contenere nelle prime righe un commento `C` che specifica il vostro Nome, Cognome e Numero di Matricola. Per esempio:

```
/*  
  Nome: Alan  
  Cognome: Turing  
  Matricola: 193700  
*/
```

File non contenenti tali informazioni NON saranno ritenuti validi. Lo script di consegna, prima di inviare la vostra soluzione al server, proverà ad eseguire il vostro codice utilizzando alcuni file di input predefiniti e controllerà

la correttezza dell'output prodotto. Tali file di input e output per la verifica del codice sono stati collocati nella cartella `dati` della vostra home directory. Dentro la cartella `dati` a loro volta i file di input e output per i test sono suddivisi in tante cartelle quanti sono gli esercizi. In questa prova abbiamo solamente l'esercizio 1, per cui dentro la cartella `dati` trovate una sola altra cartella nominata 1 dove sono presenti i file di input e output per effettuare il test dell'esercizio 1. Questi file sono nominati secondo lo schema: `input0.txt output0.txt input1.txt output1.txt input2.txt output2.txt` ... Questi file possono essere usati anche da voi per testare il vostro codice e verificare poi manualmente la correttezza dell'output prodotto. Per effettuare le vostre prove potete infatti utilizzare il comando del terminale per la redirectione dell'input. Ad esempio:

```
./compilato < dati/1/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 `dati/1/output0.txt`. Per effettuare un controllo in automatico sul primo file input `input0.txt` potete eseguire i comandi:

```
./compilato < dati/1/input0.txt > res0  
diff res0 dati/1/ouput0.txt
```

Il primo comando infatti esegue la vostra soluzione e stampa l'output prodotto nel file `res0`, il secondo infine controlla le differenze fra l'output prodotto da voi e quello corretto.

La consegna andrà a buon fine solo se il vostro codice riesce a superare tutti i test contenuti nella cartella `dati`. Eventualmente lo script di consegna vi informa per quali di questi test il vostro codice non risponde correttamente. Nel caso invece la vostra soluzione passi i test, vi verrà chiesto di specificare il vostro numero di matricola e la soluzione sarà consegnata. Una volta effettuata la prima consegna, il numero di matricola verrà associato alla macchina e tramite quella macchina non sarà possibile consegnare altri elaborati se non per quello specifico numero di matricola.

Una volta consegnata, la vostra soluzione verrà valutata nel server di consegna utilizzando altri file di test, da voi non accessibili. Il consiglio quindi quello di non provare ciecamente e ripetutamente la vostra soluzione con i file di test che vi sono forniti ma cercare di ragionare sul codice che avete scritto perché il fatto che il vostro codice passi i test di consegna non significa necessariamente che questo sia corretto in assoluto.

## Esercizio 1

Si vuole gestire la coda in un ambulatorio medico. Nell'ambulatorio è presente un solo medico e quindi i pazienti vengono visitati uno alla volta, in ordine di arrivo. Ogni paziente che arriva all'ambulatorio consegna il proprio nominativo e viene messo in coda in attesa di essere chiamato dal medico. Alla fine dell'orario di apertura dell'ambulatorio è possibile che non tutti i pazienti siano stati visitati. Si scriva quindi un programma che gestisca la coda di questo ambulatorio e alla fine dell'orario di apertura stampi in output la lista dei pazienti ancora in coda, ordinata alfabeticamente.

In particolare il programma deve entrare subito in un ciclo nel quale legge da input un intero  $e$  che codifica i seguenti eventi:

$e = 1$  Un paziente è arrivato all'ambulatorio: il programma in questo caso deve leggere un'altra volta l'input dove verrà passato il nominativo del paziente che è appena entrato e deve inserire tale nominativo in fondo alla coda. Si assuma che il nominativo del paziente sia costituito da soli caratteri alfa-numeric, non contenga spazi al suo interno e che sia lungo al massimo 100 caratteri.

$e = 2$  Il primo paziente in coda viene chiamato dal medico che lo visita. In questo caso il programma deve aggiornare la coda eliminando il primo paziente dalla coda.

$e = 0$  L'ambulatorio ha terminato l'orario di apertura, in questo caso il programma deve uscire dal ciclo.

Una volta terminato l'orario si devono stampare in output i nomi di tutti i pazienti ancora in coda in **ordine alfabetico**. In particolare, l'output deve essere formattato nel seguente modo: stampare i nominativi uno per riga e terminare la lista con una riga contenente il carattere \$ seguito da un ritorno a capo ( $\backslash n$ ). (Di conseguenza l'output richiesto per una coda *vuota* è costituito da  $\$\backslash n$ )

### Note

- Deve essere prestata particolare attenzione alla gestione dinamica della memoria. Ad esempio, si deve correttamente utilizzare la funzione `free()`.
- Non si possono fare assunzioni sulla lunghezza massima della coda.
- Per l'ordinamento, si possono sfruttare funzioni di libreria standard come `qsort` e `strcmp`.
- Se durante le vostre prove vi capitasse di non riuscire a interrompere il ciclo, per forzare la terminazione dell'esecuzione del programma si usi la combinazione di tasti `Ctrl+C`.

## Esempio

### Input

```
1
turing
1
knuth
2
2
1
lempel
1
tanenbaum
1
huffman
2
1
vonneuman
1
dijkstra
2
0
```

### Output

```
dijkstra
huffman
vonneuman
$
```