

# Prova di Laboratorio di Algoritmica

Corso A - Prof. Ferragina

Simulazione 11/05/2011

## 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. Il comando deve essere utilizzato dal terminale nel seguente modo:

```
./consegna fileSorgente.c numEx
```

dove:

- `fileSorgente.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 simulazione l'esercizio è solo il numero 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 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. 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:

```
./fileCompilato < dati/input0.txt
```

effettua il test del vostro codice sui dati contenuti nel primo file di input, assumendo che `fileCompilato` 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/output0.txt`.

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. 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 testata, in automatico, su altri file di test, da voi non accessibili. Il consiglio è 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 perchè il fatto che il vostro codice passi i test di consegna non significa necessariamente che questo sia corretto in assoluto.

**Nota:** È possibile consultare i manuali (in inglese) contenenti la spiegazione di funzionamento e la sintassi di funzioni di libreria utilizzando il comando `man`, ad esempio:

```
man qsort
```

.

## Esercizio 1

Scrivere un programma che legga da input una sequenza di stringhe, formattata nel seguente modo: la prima riga dell'input è un'intero  $N$  e poi seguono  $N$  righe in cui sono presenti stringhe alfanumeriche prive di spazi di lunghezza massima 20. Il programma deve eseguire l'ordinamento di questa sequenza di stringhe utilizzando la libreria `qsort` e implementando la seguente strategia di ordinamento:

- le stringhe di lunghezza pari devono essere posizionate nella parte superiore della sequenza mentre quelle di lunghezza dispari nella parte inferiore;
- le stringhe di lunghezza *pari*, fra di loro, devono essere posizionate in ordine alfabetico;
- le stringhe di lunghezza *dispari*, fra di loro, devono essere posizionate in ordine alfabetico inverso.

*Suggerimento:* per l'ordinamento alfabetico inverso si consiglia di utilizzare la funzione `strcmp` definita in `<string.h>` e calcolare l'inverso del risultato prodotto.