

# Algoritmica – Prova di Laboratorio

Corso A e B

Appello del 02/11/2016

## 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 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.

## Suggerimenti

**Progettare una soluzione efficiente.** Prestare attenzione ad eventuali requisiti in tempo e spazio richiesti dall'esercizio. In ogni caso, valutare la

complessità della soluzione proposta e accertarsi che sia *ragionevole*: difficilmente una soluzione con complessità  $\Theta(n^3)$  sarà accettata se esiste una soluzione semplice ed efficiente in tempo  $\mathcal{O}(n)$ .

**Abilitare i messaggi di diagnostica del compilatore.** Compilare il codice usando le opzioni `-g -Wall` di gcc:

```
gcc -Wall -g soluzione.c -o soluzione
```

risolvere *tutti* gli eventuali *warnings* restituiti dal compilatore, in particolare modo quelli relativi alle funzioni che non restituiscono un valore e ad assegnamenti tra puntatori di tipo diverso.

**Provare la propria soluzione in locale.** Valutare la correttezza della soluzione sulla propria macchina accertandosi che rispetti **tutti** gli input/output contenuti nel TestSet. Al fine di agevolare la verifica, si consiglia di posizionare la soluzione compilata e i file appartenenti al TestSet nella stessa directory e lanciare sotto tale directory lo script seguente:

```
for i in input* ; do
    ./soluzione < ${i} | diff -q - output${i##input}
done
```

**Usare Valgrind.** Nel caso in cui il programma termini in modo anomalo o non calcoli la soluzione corretta, è utile accertarsi che non acceda in modo scorretto alla memoria utilizzando **valgrind**:

```
valgrind ./soluzione < input0.txt
```

**Valgrind** eseguirà il vostro codice sull'input specificato (in questo caso, il file `input0.txt`), mostrando in output dei messaggi di diagnostica nei casi seguenti:

1. accesso (in lettura o scrittura) ad una zona di memoria non precedentemente allocata;
2. utilizzo di una variabile non inizializzata precedentemente;
3. presenza al termine dell'esecuzione del programma di zone di memoria allocate con **malloc** ma non liberate con **free** (*memory leak*).

Risolvere *tutti* i problemi ai punti 1. e 2. prima di sottoporre la soluzione al server.

## Esercizio

Scrivere un programma che legga da tastiera un array  $A$  di  $N$  interi e un intero  $K$ . Il programma deve stampare gli elementi dell'array che hanno **almeno**  $K$  occorrenze. Gli elementi devono essere stampati mantenendo il loro ordine originale in  $A$ .

La soluzione, che verrà controllata alla consegna, **deve** utilizzare una tabella hash per il conteggio delle occorrenze dei singoli elementi. In questo modo il problema può (e deve) essere risolto con due scansioni dell'array  $A$ .

L'input è formattato nel seguente modo: la prima riga contiene gli interi  $N$  e  $K$  mentre la seconda riga contiene gli elementi di  $A$  separati da uno spazio.

L'output è formato da una sola riga contenente gli elementi della soluzione separati da uno spazio.

## Esempio

### Input

```
12 3
7 2 7 3 2 7 6 5 4 4 7 4
```

### Output

```
7 7 7 4 4 7 4
```