

# Algoritmica – Esame di Laboratorio

24/01/2014

## 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 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 corrisponda a quanto contenuto nel file `output0.txt`. Per effettuare un controllo automatico sul primo file input `input0.txt` potete eseguire i comandi:

```
./compilato < input0.txt | diff - output0.txt
```

Il comando 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

Il nuovo motore di ricerca *BoING*<sup>®</sup> ha un problema con il suo servizio *Calendario* che permette ai suoi utenti di inserire eventi da ricordare. È stato deciso che *in un calendario di un utente due eventi non si possano sovrapporre*, ossia che dati due eventi, uno dei due non possa iniziare o finire all'interno dell'altro. Alcuni utenti del calendario però sono riusciti a inserire all'interno del sistema **un evento** che non rispetta questa decisione e si sovrappone con uno (**nota: uno solo**) degli altri eventi inseriti nel calendario.

L'esercizio consiste nell'identificare (se presenti) i due elementi che si sovrappongono nella lista degli eventi e eliminarli.

Un evento è rappresentato dalla struct:

```
typedef struct {
char * desc;
int start;
int end;
} event;
```

In cui **start** e **end** sono interi che rappresentano rispettivamente la data di inizio e di fine dell'evento, e **desc** è una stringa che contiene una descrizione sommaria dell'evento (ad esempio *'esame di algoritmica'*), si supponga che la stringa di descrizione non sia più lunga di 200 caratteri.

Il programma riceve in input un intero  $n$ , che rappresenta il numero di eventi nel calendario di un utente, e successivamente legge  $n$  eventi. Ogni evento è codificato su tre linee: la prima contiene la descrizione, la seconda la data di inizio **start** e la data di fine **end**.

Una volta letta la lista degli eventi, il programma deve identificare i due eventi che si sovrappongono e eliminarli (se presenti). Il programma termina stampando in output la lista delle descrizioni degli eventi che **non si sovrappongono**, ordinata lessicograficamente.

## Esempio

### Input

5  
arrampicata  
300  
400  
dentista  
100  
200  
caffè  
120  
140  
ristorante  
10  
20  
esame  
1000  
1100

dentista e caffè si sovrappongono.

### Output

arrampicata  
esame  
ristorante

Non è ammesso confrontare tutte le possibili  $O(n^2)$  coppie di eventi. Se gli eventi vengono ordinati per data di inizio crescente, è possibile che due eventi che si sovrappongono non siano adiacenti?