

# Programmazione I

## Prova di Programmazione – 5 febbraio 2015 – 2 ore 15 min

Partendo dal frammento di codice fornito, realizzare un simulatore di un *messenger* semplificato in cui due utenti, Eva e Leo, si scambiano messaggi. Il programma si limita a memorizzare la cronologia, ossia la sequenza, dei messaggi inseriti dai due utenti. Le lunghezze massime di ciascun messaggio e della cronologia (ossia il numero massimo di messaggi) sono definiti a tempo di scrittura del programma. All'avvio del programma la cronologia è vuota. Il programma deve fornire le seguenti funzionalità.

1. **scrivi\_messaggio(utente)** Legge una riga da *stdin*, assumendo che sia il messaggio scritto dall'utente **utente** (che può essere Eva oppure Leo), ed aggiunge tale messaggio in fondo alla cronologia. Assumere che l'utente: 1) inserisca parole non più lunghe di 18 caratteri, 2) usi solo il carattere spazio come spazio bianco, 3) inserisca un solo spazio tra le parole e 4) non inserisca uno spazio né come primo né come ultimo carattere del messaggio. Fare riferimento al punto 5 per un esempio di cronologia ottenuta mediante una serie di invocazioni di questa funzionalità (trascurare la particolare formattazione richiesta in tale punto).
2. **salva\_messaggi** Salva la cronologia in un file di testo.
3. **carica\_messaggi** Carica la cronologia dal file. La precedente cronologia è persa.
4. **lunghezza\_parola(stringa, i)** Possibile funzionalità di supporto per la prossima funzionalità: ritorna la lunghezza della parola presente nella stringa **stringa** passata in ingresso, a partire dalla posizione **i**-esima. Ritorna 0 se in tale posizione vi è uno spazio o il terminatore.
5. **stampa\_messaggi** Stampa la cronologia col seguente formato. Stampa sia i messaggi di Eva che i messaggi di Leo in una colonna di testo di larghezza pari a 20 caratteri. Per Eva la colonna è allineata a sinistra, mentre per Leo è allineata a destra. In particolare, la colonna per Leo va stampata ad un carattere di distanza dalla posizione in cui la colonna per Eva termina. Si ottiene il voto massimo se il testo è giustificato a sinistra e le parole non vengono spezzate per andare a capo. Ad esempio (i numeri stampati nella prima riga servono solo a mostrarvi la larghezza e la posizione delle due colonne):

```
12345678901234567890 12345678901234567890
```

```
Ciao
```

```
Come va?
```

```
Ciao
```

```
Come e' andata la  
prova scritta?
```

```
Superata, ora sto  
svolgendo la prova  
pratica
```

Notare che Eva scrive due messaggi di seguito prima che Leo scriva il suo primo messaggio. Implementare la stampa di ciascun messaggio con una funzione a cui si passa il testo del messaggio da stampare ed eventuali altri parametri.

I parametri di ingresso delle funzionalità sono solo indicativi. Gestire opportunamente le situazioni di errore, tranne l'inserimento di dati in formato errato e di messaggi troppo lunghi da *stdin*.

---

### REGOLE

- Si può utilizzare ogni genere di manuale o di materiale didattico di altra natura
- Per superare la prova, il programma deve essere perfettamente funzionante nelle parti 1 e 2. Il voto ottenuto in questo caso è 18.
- Ciascuna funzionalità DEVE essere implementata mediante almeno una funzione.
- Il voto massimo (almeno 30) si ottiene se
  - a) il programma è perfettamente funzionante in ogni sua parte
  - b) tutti i principi di ingegneria del codice visti nel corso sono stati applicati