

Programmazione I

Prova di Programmazione – 14 febbraio 2013 – 2 ore

Partendo dal frammento di codice fornito, realizzare un programma che gestisce un insieme di contatori di generiche *richieste*, in cui ogni contatore è caratterizzato dal nome del tipo di richieste (una parola la cui lunghezza non è nota a tempo di scrittura del programma) e da un numero intero che costituisce il contatore vero e proprio. Il numero massimo di contatori è noto a priori. All'avvio del programma l'insieme di contatori è vuoto. Realizzare le seguenti funzionalità.

1. **aggiungi_contatore(nome)** Aggiunge un contatore di nome **nome**, inizializzato a 0.
2. **incrementa_contatore(i)** Incrementa il contatore di indice **i**.
3. **stampa_contatori** Stampa tipo di richiesta e valore di ciascun contatore. Ad esempio:

```
Richieste-A 2
Letture      0
Richieste-F 19
```
4. **salva_contatori** Salva l'insieme di contatori in un file di testo dal nome predefinito.
5. **carica_contatori** Carica l'insieme dal file. Il precedente contenuto dell'insieme è perso.
6. **redireziona(i1, i2)** Redireziona gli incrementi del contatore di indice **i1** verso il contatore di indice **i2**. Se **i1 == i2**, si elimina il redirezionamento per il contatore di indice **i1**. Il redirezionamento è utilizzato nel prossimo punto.
7. **incrementa_contatore2(i)** Incrementa il contatore di indice **i** tenendo conto dei redirezionamenti. Ad esempio, se si prova ad incrementare un contatore di indice *i1*, che è redirezionato verso un contatore di indice *i2*, incrementa il contatore di indice *i2* e non quello di indice *i1*. Se però anche il contatore di indice *i2* fosse stato redirezionato verso un contatore di indice *i3*, allora si incrementerebbe il contatore di indice *i3* e non quello di indice *i2*. E così via.

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

REGOLE

- Si può utilizzare ogni genere di manuale o di altro materiale didattico
- Per superare la prova, il programma deve essere perfettamente funzionante nelle parti 1, 2 e 3. 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