

# Programmazione I

## Prova di Programmazione – 11 gennaio 2012 – 2 ore 15 min

Un *line editor* permette di modificare documenti di testo lavorando su una riga alla volta. Consideriamo un semplice *line editor* che lavora su di un solo documento alla volta, con il documento caratterizzato a sua volta da un numero massimo di righe e da una lunghezza massima di ciascuna riga noti a tempo di scrittura del programma. All'inizio il documento corrente è vuoto. Partendo dal frammento di codice fornito, scrivere un programma che realizza tale editor attraverso le seguenti funzionalità:

1. **modifica\_aggiungi\_riga(idx)** Legge una riga da *stdin* (senza inserire nella riga l'eventuale *newline* presente su *stdin* in fondo alla riga stessa) e, se tale riga contiene almeno un carattere, sostituisce la riga di indice **idx** del documento con la riga letta da *stdin*. Se **idx** è maggiore dell'indice dell'ultima riga del documento, aggiunge la riga letta da *stdin* in fondo al documento.
2. **stampa\_documento** Stampa il contenuto del documento, una riga dopo l'altra.
3. **salva\_documento** Salva il documento in un file di testo dal nome predefinito.
4. **carica\_documento** Carica il documento da un file di testo. Il precedente contenuto è perso.
5. **elimina\_riga(idx)** Elimina la riga di indice **idx** dal documento.
6. **cerca\_prossima\_occorrenza(p, idx, pos)** Cerca la prossima occorrenza della parola **p** nella porzione di documento che va dalla posizione **pos** nella riga di indice **idx** (una parola è una sequenza di caratteri non separati da spazi). In caso di successo, ritorna l'indice della riga e la posizione nella riga in cui l'occorrenza è stata trovata.

Consideriamo, ad esempio, il seguente documento di tre righe (le prime due righe seguenti non fanno parte del documento ma servono solo da riferimento per determinare la posizione di ciascuna parola all'interno delle righe del documento)

```
0           1           2           3           4           5           6
0123456789012345678901234567890123456789012345678901234567890
Un programma fa quello che gli dici, non quello che pensi.
La complessita' di un programma si arresta solo dopo aver
oltrepassato le capacita' del programmatore.
```

Se invocata con parametri (*quello*, 0, 0), la funzionalità ritorna la coppia di valori (0, 16). Se invocata invece con parametri (*quello*, 0, 18) ritorna (0, 41), con (*programma*, 0, 20) ritorna (1, 22), infine con (*programma*, 1, 25) ritorna (2, 30).

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