

# Programmazione I

## Prova di Programmazione – 25 Giugno 2016 – 2 ore

Partendo dal frammento di codice fornito, realizzare un programma per la gestione dello stato di un display da stadio composto da N righe di M pixel. Ogni pixel può essere in uno solo di due stati: OFF (spento) e ON (acceso). Si assuma che il numero di righe N sia un valore costante (es., 5), definito a tempo di scrittura del programma e immutabile. Il numero di pixel in ciascuna riga (uguale per tutte le righe) è invece un parametro che può variare nel corso dell'esecuzione del programma. All'avvio del programma la sequenza che descrive ciascuna riga del display sia inizializzato a un numero di pixel definito a tempo di scrittura del programma (es., 5). I pixel sono inizialmente tutti accesi. Il programma deve fornire le seguenti funzionalità.

1. **modifica\_stato\_pixel (riga, num\_pixel, nuovo\_stato)** Modifica in *nuovo\_stato* (ON, OFF) lo stato del pixel di indice *num\_pixel* nella struttura che descrive la *riga* di interesse.
2. **stampa\_display** Stampa l'intero display, ovvero lo stato di ciascuna delle N sequenze di M pixel. Si rappresentino con un \* i pixel ON e con un \_ quelli OFF. Ad esempio, per un display NxM=4x4 l'output potrebbe essere:

```
****
*  _
*  _
****
```

3. **modifica\_numero\_pixel\_per\_riga (nuovo\_numero)** Modifica in *nuovo\_numero* la lunghezza di ciascuna riga del display. *nuovo\_numero* deve essere maggiore di zero. Supponendo che la precedente lunghezza fosse M, inizializza lo stato dei pixel della nuova sequenza nel modo seguente. Se la sequenza viene estesa, allora lo stato dei primi M pixel della sequenza estesa deve essere uguale a quello degli M pixel della sequenza originale, mentre i restanti pixel nella sequenza estesa devono essere tutti nello stato ON. Se invece la sequenza viene accorciata, allora lo stato dei *nuovo\_numero* pixel della sequenza ridotta deve essere uguale allo stato dei primi *nuovo\_numero* pixel della sequenza originale.
4. **salva\_display** Salva lo stato del display, ovvero lo stato di tutte le sequenze di pixel (righe) in un file di testo dal nome definito a tempo di scrittura del programma (*display.txt*).
5. **carica\_display** Carica lo stato del display, ovvero lo stato di tutte le sequenze di pixel (righe) dal file. Lo stato precedente è perso.
6. **modifica\_stato\_riga (sequenza\_stati)** Modifica lo stato dei primi M pixel di una riga del display, ove M è la lunghezza della stringa *sequenza\_stati*. La stringa *sequenza\_stati* deve essere uguale ad una sequenza di \* e \_. In particolare, per ogni \* nella posizione i-esima all'interno della stringa *sequenza\_stati*, questa funzionalità imposta a ON lo stato del pixel i-esimo, mentre per ogni \_ nella posizione i-esima imposta a OFF lo stato del pixel i-esimo.
7. **modifica\_stato\_display** Modifica lo stato del display, leggendo da stdin lo stato di ogni riga.

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 stringhe troppo lunghe 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