

**PROGRAMMAZIONE I (A,B) - a.a. 2017-18**  
**V appello – 6 Settembre 2018**

**Esercizio 1**

Dato il seguente linguaggio sull'alfabeto  $\Sigma = \{a, b, c\}$

$$L = \{a^n b^m c^k \mid n, m > 0 \wedge k = (n + m) \% 3\}$$

si verifichi se tale linguaggio è regolare o meno e si dia una grammatica che lo generi.

**Esercizio 2**

Si scriva una funzione **C** che, dato un array *a* di dimensione *dim<sub>a</sub>* e un array *b* di dimensione *dim<sub>b</sub>*, restituisca il risultato della seguente formula:

$$\begin{aligned} & \#\{i \mid i \in [0, \text{dim}_a) \wedge \neg(\exists j \in [0, \text{dim}_b). a[i] = b[j])\} \\ & + \#\{i \mid i \in [0, \text{dim}_b) \wedge \neg(\exists j \in [0, \text{dim}_a). b[i] = a[j])\} \end{aligned}$$

**Esercizio 3**

Si definisca in CAML, senza usare la ricorsione esplicita, una funzione

```
init : 'a list -> 'a list -> 'a list
```

che, date due liste **lis1** e **lis2**, restituisce la lista degli elementi di **lis1** che precedono il primo elemento di tale lista che è contenuto anche in **lis2**. Gli elementi nella lista restituita devono rispettare l'ordine che avevano in **lis1**. La funzione restituisce una lista vuota se nessun elemento di **lis1** è contenuto in **lis2**.

**Esercizio 4**

Si definisca in CAML, usando la ricorsione esplicita, funzione **init** descritta nell'Esercizio 3, con la differenza che in questo caso se nessun elemento di **lis1** è contenuto in **lis2**, la funzione deve restituire l'intera lista **lis1**.