

**PROGRAMMAZIONE I (A,B) - a.a. 2017-18**  
**IV appello – 16 Luglio 2018**

**Esercizio 1**

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

$$L = \{a^n b^m c^k \mid n, m, k > 0 \wedge (nmk) \% 2 = 0\}$$

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

**Esercizio 2**

Si scriva una funzione **C** che, dato un array  $a$  di dimensione  $dim$  e un intero  $k$ , restituisca il valore di verità della seguente formula:

$$\exists n \in [0, dim). (n \leq \min(k, dim - 1 - k) \wedge \sum_{i=k-n}^{k+n} a[i] = 0)$$

dove  $\min(x, y)$  rappresenta il minimo tra  $x$  e  $y$ .

**Esercizio 3**

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

```
listasomme : int list -> int list
```

che, data una lista di interi, restituisce la lista delle somme degli elementi che si trovano tra due occorrenze del valore 0. Ad esempio: `listasomme [1;2;0;3;4;0;5;0;8] = [7; 5]`, `listasomme [1;2;3;0]=[]`, `listasomme [2;3]=[]`, `listasomme [0;2;3;0;4]=[5]`.

**Esercizio 4**

Si definisca in CAML, usando la ricorsione esplicita, una funzione

```
inizio : 'a list -> 'a -> 'a list
```

che, data una lista `lis` e un elemento `n`, restituisce la lista degli elementi che compaiono prima della prima occorrenza di `n`. Se `n` non è presente, restituisce `[]`.