

PROGRAMMAZIONE I (A,B) - a.a. 2016-17

Quarto Appello – 11 Luglio 2017

Esercizio 1

Si scriva una funzione **C** che, dato un array a di dimensione dim_a e un array b di dimensione dim_b , restituisce il valore di verità della seguente formula:

$$\forall i \in [0, dim_a). \exists j \in [0, dim_b). ((a[i] = b[j]) \wedge (\forall k \in [0, dim_b). ((k \neq j) \Rightarrow (b[j] \neq b[k])))$$

Esercizio 2

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

$$\mathcal{L} = \{a^n b^{2m} \mid n > 0 \wedge n \% 3 = 1 \wedge m > 0\}$$

si verifichi se il linguaggio è regolare o meno (fornendo una opportuna dimostrazione) e si definisca una grammatica che lo genera.

Esercizio 3

Si suppongano predefiniti i tipi

```
struct el {int info; struct el *next;};
typedef struct el ElementoDiLista;
typedef ElementoDiLista* ListaDiElementi;
```

Si scriva in **C** una procedura che, presa una lista, elimina da essa tutti gli elementi compresi tra il primo elemento che contiene il valore 0 e l'ultimo elemento che contiene il valore 0. Tali due elementi contenenti 0 non devono essere eliminati. Se la lista contiene meno di due elementi contenenti il valore 0, nessuna eliminazione dovrà essere eseguita.

Esercizio 4

Si definisca in CAML una funzione

```
contafrequente : int list -> int
```

che, data una lista lis di interi, calcola il numero di occorrenze dell'elemento che occorre il maggior numero di volte nella lista. La funzione può essere liberamente definita facendo uso o meno della ricorsione esplicita.