

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

Terzo Appello – 15 Giugno 2017

Esercizio 1

Si scriva una funzione **C** che, dati due array a e b , entrambi di dimensione dim , verifica la seguente proprietà:

- ogni elemento contenuto nell'array a in una posizione con indice pari è presente anche nell'array b (in una posizione qualunque)
- ogni elemento contenuto nell'array a in una posizione con indice dispari è minore di tutti gli elementi contenuti in b

La funzione deve restituire 1 se la proprietà è verificata, 0 altrimenti.

Esercizio 2

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

$$\mathcal{L} = \{ca^n cb^m c \mid n < 2m\}$$

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 e un intero x , ridispone gli elementi della lista in ordine inverso rimuovendo al contempo tutti gli elementi che contengono il valore x .

Esercizio 4

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

```
sommadopozero : int list -> int
```

che, data una lista `lis` di interi, calcola la somma di tutti gli elementi che seguono la prima occorrenza del valore 0 nella lista. Se la lista non contiene nessuno 0, il risultato calcolato dalla funzione deve essere 0. La funzione deve essere definita senza rovesciare la lista.