

# PROGRAMMAZIONE 1 e LABORATORIO (A,B)

a.a. 2010-2011

## Prova scritta del 1 luglio 2011

Scrivere IN STAMPATELLO COGNOME, NOME, MATRICOLA e CORSO su ogni foglio consegnato

### ESERCIZIO 1 (punti 5)

Dato l'alfabeto  $\Lambda = \{0, 1\}$  si definisca una grammatica che genera il seguente linguaggio su  $\Lambda^+$

$$\{\alpha \in \Lambda^+ \mid \|\alpha\|_0 = \|\alpha\|_1\}$$

dove la notazione  $\|\alpha\|_x$  indica il numero di occorrenze del simbolo  $x$  nella stringa  $\alpha$ .

### ESERCIZIO 2 (punti 5)

Un albero binario si dice *perfettamente bilanciato* se e solo se, per ogni sottoalbero  $bt$ , la profondità del sottoalbero sinistro di  $bt$  è uguale alla profondità del sottoalbero destro di  $bt$ .

Dato il tipo degli alberi binari visto a lezione

```
type 'a btree = Void | Node of 'a * 'a btree * 'a btree
```

si definisca una funzione

```
bilanciato : 'a btree -> bool
```

in modo che `bilanciato bt` restituisca *true* se `bt` è perfettamente bilanciato, *false* altrimenti. Sono consentiti la definizione e l'uso di funzioni ausiliarie.

### ESERCIZIO 3 (punti 5)

Scrivere una procedura C che, dato un array  $a$  di dimensione  $dim$ , azzeri tutti gli elementi  $a[i]$  per i quali vale la seguente proprietà

$$\#\{j \in [0, i) \mid a[j] = 0\} = 2.$$

Si ricorda che la notazione  $\#S$  indica la cardinalità dell'insieme  $S$ . Dato ad esempio il seguente array

3	0	0	2	6	0	5
---	---	---	---	---	---	---

la procedura lo deve modificare come segue

3	0	0	0	0	0	5
---	---	---	---	---	---	---

### ESERCIZIO 4 (punti 5)

Senza utilizzare ricorsione esplicita, ma solo funzioni di ordine superiore, si definisca in CAML una funzione

```
splitposneg : int list -> (int list * int list)
```

in modo che se `splitposneg lis = (l1, l2)`, `l1` contenga tutti e soli gli elementi non negativi di `lis` e `l2` contenga tutti e soli gli elementi negativi di `lis`.

### ESERCIZIO 5 (punti 5)

Scrivere una funzione C che, dato un array  $a$  di dimensione  $dim$ , restituisce 1 se la sequenza  $a[0]a[1] \dots a[n-1]$  appartiene al linguaggio dell'esercizio 1, restituisce 0 altrimenti.

### ESERCIZIO 6 (punti 5)

Si suppongano date le seguenti definizioni:

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

Scrivere in C una procedura che, data una lista, cancella l'ultimo elemento se e solo se contiene un valore uguale al penultimo. La funzione deve lasciare invariata la lista se contiene meno di due elementi.