

# PROGRAMMAZIONE 1 e LABORATORIO (A,B) - a.a. 2011/12

## Prova scritta del 4 giugno 2012

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

Negli esercizi di programmazione in C si assuma predefinito il tipo `typedef enum {false, true} boolean`.

### ESERCIZIO 1 (punti 6)

Un albero binario si dice *quasi perfettamente bilanciato* se e solo se:

- è vuoto, oppure
- non è vuoto, il sottoalbero destro e il sottoalbero sinistro hanno profondità che differiscono al più di 1 ed entrambi sono quasi perfettamente bilanciati.

Dato il tipo degli alberi binari visto a lezione

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

si definisca in CAML una funzione che controlla se un albero binario è quasi perfettamente bilanciato.

### ESERCIZIO 2 (punti 6)

Si definisca in C una funzione

```
boolean check (int a[], int b[], int dim)
```

che restituisce il valore di verità della seguente formula:

$$(\exists j \in [0, dim). (\forall i \in [0, dim - j). a[i] = b[i + j]) \wedge (\forall i \in [dim - j, dim). a[i] = b[i - (dim - j)]))$$

Suggerimento: si cerchi di comprendere il significato della proprietà espressa dalla formula.

### ESERCIZIO 3 (punti 6)

Si consideri la seguente grammatica  $G = \langle \{1, 2, 3\}, \{S, A, B\}, S, P \rangle$ , dove le produzioni in  $P$  sono le seguenti:

```
S ::= S3 | A
```

```
A ::= 1A | B
```

```
B ::= 2B | 2
```

Scrivere in C una funzione che, dato un array di interi  $a$  e la sua dimensione  $dim$  restituisce `true` se la sequenza  $a[0]a[1] \dots a[dim-1]$  appartiene al linguaggio generato da  $G$ , e restituisce `false` altrimenti.

### ESERCIZIO 4 (punti 6)

Si completi la seguente definizione

```
let tris l = let f x y = ... in foldr f ... l
```

in modo che

```
tris : int list -> int * int list * int list
```

e `(tris lis)` restituisca la tripla  $(s, l1, l2)$ , in cui  $s$  è la somma degli elementi di  $lis$ ,  $l1$  è la lista di tutti gli elementi di  $lis$  che non sono maggiori della somma degli elementi che li seguono in  $lis$ , e  $l2$  è la lista di tutti gli elementi di  $lis$  che sono maggiori della somma degli elementi che li seguono in  $lis$ .

### ESERCIZIO 5 (punti 6)

Date le seguenti definizioni:

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

scrivere in C una procedura che, dati in ingresso attraverso opportuni parametri un intero  $x$  e una lista di interi, con la proprietà che tutti gli elementi negativi precedono quelli non negativi, inserisce  $x$  nella lista in modo da preservare la stessa proprietà.