

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

## Prova scritta del 31 gennaio 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)

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 `check` con tipo

```
check : 'a btree -> 'a -> int -> bool
```

in modo che `(check bt x n)` restituisca `true` se l'albero `bt` contiene una foglia etichettata `x` a profondità `n`, restituisca `false` altrimenti.

### ESERCIZIO 2 (punti 6)

Si definisca in C una funzione

```
boolean twice (int a[], int dima, int b[], int dimb)
```

che restituisce il valore di verità della seguente formula:

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

### ESERCIZIO 3 (punti 6)

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

```
S ::= 1S | 1D
```

```
D ::= 2D | 2T
```

```
T ::= 3T | 3A
```

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

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

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]...a[dim-1]` appartiene al linguaggio generato da  $G$ , e restituisce `false` altrimenti.

### ESERCIZIO 4 (punti 6)

Si completi la seguente definizione

```
let foo n l = let f x y = ... in foldr f ... l
```

in modo che `(foo z lis)` restituisca la coppia `(num, sum)`, in cui `num` è il numero di elementi di `lis` maggiori di `z` e `sum` è la somma di tali elementi.

### 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 una lista di interi `lista1`, un intero `x` e una seconda lista di interi `lista2`, modifica `lista1` concatenando ad essa un nuovo elemento contenente il valore `x`, seguito dalla lista `lista2`.