

# PROGRAMMAZIONE 1 e LABORATORIO (A,B) - a.a. 2014-2015

## Prova scritta del 1/09/2015

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

### ESERCIZIO 1 (6 punti)

Si dimostri, utilizzando il pumping lemma, che il seguente linguaggio sull'alfabeto  $\{a, b\}$  non è regolare

$$\mathcal{L} = \{ab^n ab^{n+1} \mid n > 0\}$$

### ESERCIZIO 2 (6 punti)

Dato il tipo degli alberi binari visti a lezione

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

si scriva in CAML una funzione

```
foo : 'a btree -> 'a list * 'a list
```

che, dato un albero binario, restituisce una coppia di liste  $(l1, l2)$  in cui  $l1$  contiene tutti i valori dei nodi dell'albero con esattamente due figli e  $l2$  tutti i valori degli altri nodi dell'albero. Ad esempio

```
foo(Node(1, Node(2, Void, Void), Node(4, Node(3, Void, Void), Node(1, Void, Void)))) = [1;4], [2;3;1]
```

### ESERCIZIO 3 (6 punti)

Si definisca in CAML, senza utilizzare ricorsione esplicita, una funzione

```
foo : ('a -> bool) -> 'a list -> 'a list
```

in modo che  $(foo\ p\ xs)$  restituisca una lista che contiene tutti gli elementi di  $xs$  e in cui gli elementi che soddisfano  $p$  precedono gli elementi che non soddisfano  $p$ .

### ESERCIZIO 4 (6 punti)

Si scriva in C una procedura che, presi attraverso opportuni parametri una lista di interi e un intero  $x$ , aggiunge un elemento che contiene  $x$  prima di ogni elemento negativo presente nella lista.

Si suppongano predefiniti i tipi

```
struct el {int info; struct el *next;};
```

```
typedef struct el ElementoDiLista;
```

```
typedef ElementoDiLista *ListaDiInteri;
```

### ESERCIZIO 5 (6 punti)

Si scriva in C una funzione che, dato un array di interi  $a$ , la sua dimensione  $dim_a$  ed un intero  $R$ , restituisce il valore di verità della seguente formula

$$(\forall i, j \in [0, dim_a). j > i \wedge a[i] = a[j] \implies j - i = R)$$