

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

## Prova scritta del 23/06/2015

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  $\{a, b, c\}$  non è regolare

$$\mathcal{L} = \{a^n b^k c^m \mid n, k, m > 0 \wedge (n = k \vee n = m)\}$$

### ESERCIZIO 2 (6 punti)

Dato un albero binario, si definisce *livello* di un nodo nell'albero il numero di nodi che si incontrano nel cammino dalla radice al nodo medesimo. Quindi, ad esempio, la radice ha livello 1, i figli della radice hanno livello 2, e così via. 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

```
even : 'a btree -> 'a list
```

che, dato un albero binario, restituisce la lista con i valori di tutti i nodi che occorrono a livelli pari nell'albero.

### ESERCIZIO 3 (6 punti)

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

```
split : 'a list -> 'a list list
```

in modo che `(split xs)` restituisca la lista delle più lunghe sottoliste non decrescenti in `xs`.

Ad esempio

```
split [1;2;2;1;5;4;6;3;2;3] = [[1;2;2]; [1;5]; [4;6]; [3]; [2;3]]
```

### ESERCIZIO 4 (6 punti)

Si scriva in C una procedura che, prese attraverso opportuni parametri due liste di interi, aggiunge alla seconda lista tutti gli elementi della prima che non appartengono anche alla seconda.

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, dati due array di interi `a` e `b` e le loro dimensioni `dima` e `dimb`, restituisce il valore

$$\min I$$

dove

$$I = \{i \mid i \in [0, dima) \wedge \#\{j \mid j \in [0, dimb) \wedge a[i] = b[j]\} \bmod 2 = 0\}$$

se `I` non è vuoto; restituisce `dima` se `I` è vuoto.

Si ricorda che:

- dato un insieme finito  $A$ ,  $\#A$  indica il numero di elementi di  $A$
- dato un insieme finito non vuoto di numeri  $S$ ,  $\min S$  indica il valore minimo dell'insieme.