

# PROGRAMMAZIONE 1 e LABORATORIO (A,B) - a.a. 2012/13

## Prova scritta del 15 luglio 2013

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

### ESERCIZIO 1 (punti 5)

Si definisca una grammatica libera che, dato l'alfabeto  $\Lambda = \{a, b, c\}$  genera il linguaggio formato da tutte e sole le stringhe di  $\Lambda^*$  in cui il numero di occorrenze del simbolo  $a$  è maggiore di 0 e minore di 3.

### ESERCIZIO 2 (punti 5)

Date le seguenti definizioni di tipo:

```
type 'a Pair = Pair of 'a * 'a
type 'a Pairs = Empty | Seq of 'a Pairs * 'a Pair
```

definire una funzione

```
count_same : 'a Pairs -> int
```

in modo che `(count_same ps)` restituisca il numero di coppie in `ps` in cui il primo ed il secondo elemento sono uguali.

### ESERCIZIO 3 (punti 5)

Scrivere una funzione `C` che, dati due array di interi e le loro dimensioni, controlla che ciascuno contenga almeno un elemento non presente nell'altro.

### ESERCIZIO 4 (punti 5)

Senza utilizzare ricorsione esplicita, ma solo funzioni di ordine superiore, si definisca in `CAML` una funzione che, data una lista di interi, restituisce una coppia formata dall'elemento minimo della lista e dal numero di elementi che precedono la prima occorrenza del minimo. Ad esempio, `foo [10;0;-1;3;5;6;-1;8] = (-1,2)`. La funzione non è definita se la lista è vuota.

### ESERCIZIO 5 (punti 5)

Date le seguenti definizioni:

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

scrivere in `C` una procedura che, data in ingresso una lista di interi, la modifica in modo che ciascun elemento della lista occorra in essa una e una sola volta.

### ESERCIZIO 6 (punti 5)

Si definisca una procedura `C` che, dato un array di interi e la sua dimensione, raddoppia tutte le occorrenze del primo elemento che compare nell'array almeno due volte. Ad esempio, dato l'array

6	2	6	2	10	6	2	2
---	---	---	---	----	---	---	---

la procedura lo deve modificare come segue

12	2	12	2	10	12	2	2
----	---	----	---	----	----	---	---