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

Prova scritta del 5 settembre 2013

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

ESERCIZIO 1 (punti 5)
Si completi, definendo gli insiemi V e P, la grammatica

$$G = \{\{0,1,2,\ldots,9,\ ;\ ,\ -\ ,\ [\ ,\ ]\ \}, V \cup \{Int, List\}, List, P \cup P_{Int}\}$$

in modo che generi tutte le costanti di tipo `int list` di `CAML`.
Si supponga di avere già a disposizione le produzioni P_{Int} che generano i numeri interi a partire dalla categoria sintattica Int.

ESERCIZIO 2 (punti 5)
Dato il tipo sequenza

```
type 'a sequenza = Info of 'a * Next of 'a sequenza | Null
```

si scriva una funzione `CAML`

```
tolist: 'a sequenza -> 'a list
```

che trasforma una `'a sequenza` s in una lista che contiene tutti i valori di tipo `'a` in s.

ESERCIZIO 3 (punti 5)
Scrivere una funzione `C` che, dato un array di interi e la sua dimensione, controlla che gli elementi con indice dispari siano tutti distinti fra loro.

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 almeno due elementi, porta in prima posizione il penultimo elemento.
N.B.: la funzione non è definita su liste vuote o con un solo elemento.

ESERCIZIO 5 (punti 5)
Date le seguenti definizioni:

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

scrivere in `C` una procedura che, data una lista, porta in seconda posizione l'ultimo elemento. La lista deve rimanere inalterata se è vuota o contiene un solo elemento.
N.B.: la procedura non deve utilizzare operazioni di assegnamento sui campi `info` degli elementi della lista, il cui tipo `T` non è ulteriormente specificato.

ESERCIZIO 6 (punti 5)
Un array a di dimensione $[0, dima)$ *estende* un array b di dimensione $[0, dimb)$ se e solo se

$$\forall i \in [0, dima).\ \#\{j \mid j \in [0, dima) \wedge a[i] = a[j]\} = \ \#\{k \mid k \in [0, dimb) \wedge a[i] = b[k]\} + 1$$

(si ricorda che $\#A$ indica la cardinalità, ovvero il numero di elementi, dell'insieme A).
Scrivere in `C` una funzione

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

che restituisce `true` se a estende b, restituisce `false` altrimenti.
(Si supponga predefinito il tipo `boolean`).