PROGRAMMAZIONE 1 e LABORATORIO (A,B) - a.a. 2014-2015
Prova scritta del 3/06/2015

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

$$\mathcal{L} = \left\{ a^n b^m c^k \;\mid\; n > m + k \geq 1 \right\}$$

sull'alfabeto $\Lambda = \{a, b, c\}$ non è regolare.

ESERCIZIO 2 (6 punti)
Si supponga di disporre di una griglia 2×2 e di una pedina sulla griglia. La pedina si può muovere in verticale verso l'alto ($\uparrow$) o verso il basso ($\downarrow$), oppure in orizzontale verso sinistra ($\leftarrow$) o verso destra, ($\rightarrow$). La pedina non può mai uscire dai limiti della griglia.
Si dia una grammatica sull'alfabeto $\Lambda = \{\uparrow, \downarrow, \leftarrow, \rightarrow\}$ in modo che le stringhe generate siano tutte e sole quelle che rappresentano sequenze di movimenti che portano la pedina dalla prima casella in basso a sinistra alla stessa casella (al termine della sequenza).
La grammatica deve generare, ad esempio, le sequenze $\uparrow\rightarrow\leftarrow\rightarrow\downarrow\leftarrow$ e $\rightarrow\uparrow\downarrow\leftarrow\uparrow\downarrow$, ma non deve generare, ad esempio, le sequenze $\uparrow\rightarrow\leftarrow\rightarrow$ e $\rightarrow\uparrow\downarrow$.

ESERCIZIO 3 (6 punti)
Si definisca in CAML, senza utilizzare ricorsione esplicita, una funzione

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

in modo che `(split xxs)` restituisca la coppia `(ys, zs)` in cui

- se `xxs` contiene almeno una occorrenza della lista vuota, `ys` è la concatenzione delle liste in `xxs` che precedono l'ultima occorrenza della lista vuota e `zs` è la concatenzione delle liste in `xxs` che seguono l'ultima occorrenza della lista vuota;
- se `xxs` non contiene la lista vuota, `ys=[]` e `zs=xxs`.

ESERCIZIO 4 (6 punti)
Si scriva in `C` una procedura che, dati attraverso opportuni parametri una lista di interi e un intero $n \geq 0$, elimina dalla lista i primi n elementi positivi. Si suppongano predefiniti i tipi

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

typedef struct el ElementoDiLista;
typedef ElementoDiLista *ListaDiInteri;
```

ESERCIZIO 5 (6 punti)
Dato un intero x e un array di interi v di dimensione dim, denotiamo con $x \circledcirc v$ la seguente formula

$$x \circledcirc v \equiv (\exists i.\ 0 \leq i < dim \wedge x = v[i])$$

Si scriva una funzione `C`

```
int check (int a[], int b[], int dima, int dimb)
```

che, dati due array `a` e `b` e le loro dimensioni `dima` e `dimb`, restituisce il valore di verità della seguente formula

$$\exists i \in [0, \texttt{dima}).\ \big((\forall j \in [0, i).\ \texttt{a}[j] \circledcirc \texttt{b}) \wedge \neg(\exists j \in [i, \texttt{dima}).\ \texttt{a}[j] \circledcirc \texttt{b})\big)$$