Fondamenti di Programmazione - CdL in MATEMATICA

I appello del 26/6/2009

NON è consentito utilizzare comandi che provocano forzatamente l'uscita dai cicli, né è consentito l'uso di variabili **globali**.
Se necessario, si utilizzi il tipo `boolean` definito da `typedef enum {false, true} boolean`.

n. eserc.	1	2	3	4
punt. tot	7	7	6	12
punt. ott.				

ESERCIZIO 1 (7 punti)

- Costruire, motivandolo informalmente, un DFA che riconosca le stringhe binarie che rappresentano numeri naturali multipli di 3.
- Trovare l'espressione regolare corrispondente.
- Trovare la grammatica **regolare** corrispondente.
- Costruire, motivandolo informalmente, un DFA che riconosca le stringhe binarie che rappresentano numeri naturali che siano sia multipli di 3 che dispari.

ESERCIZIO 2 (7 punti)
Definire, in modo **ricorsivo**, una funzione con prototipo

```
boolean ControllaOcc(int *v1, int dim, int val, int occ)
```

che, dato un array di interi, controlla se esistono **esattamente** *occ* occorrenze del valore *val*.
Ad esempio, dato il vettore v_1

6	2	8	9	4	9	3	1	9	16

dobbiamo avere che `ControllaOcc(v1, 10, 9, 3)` = `true`, mentre `ControllaOcc(v1, 10, 9, 2)` = `false`.

ESERCIZIO 3 (6 punti)
Data una funzione con prototipo

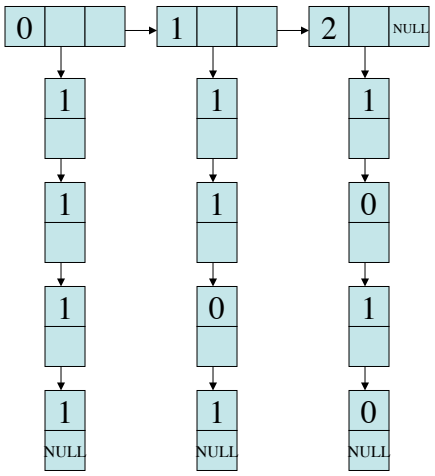
```
boolean P(int a, int b)
```

definire una funzione con prototipo

```
boolean ExistsForall(int *v1, int *v2, int dim)
```

tale che $\exists i, \forall j.\ P(a[i], b[j])$, con $i, j \in [0, dim - 1]$.

ESERCIZIO 4 (12 punti)
Si vuole rappresentare un sistema di *car pooling* o *car sharing*. Vogliamo quindi avere una lista concatenata, dove ogni elemento rappresenta una vettura, con il suo numero di posti liberi e la sua sottolista dei posti in dotazione alla vettura. La sottolista dei posti in dotazione ci dice per ogni posto se è libero oppure no. Di seguito riportiamo un esempio.



- (1 punti) Definire i tipi opportuni per la rappresentazione indicata.
- (5 punti) Scrivere una funzione *iterativa* che controlli se esistono n posti liberi nella stessa vettura e che restituisca il puntatore al primo posto libero della vettura individuata, oppure `NULL`.
- (6 punti) Scrivere una procedura *ricorsiva* che permetta di inserire una nuova vettura, con s posti disponibili, subito dopo la prima vettura satura (ovvero con 0 posti disponibili). Se non esistono vetture sature, la nuova vettura va collocata alla fine della lista.