

FONDAMENTI DI PROGRAMMAZIONE - CdL in MATEMATICA

PROVA SCRITTA DEL 20/6/2011

Scrivere **in stampatello** COGNOME, NOME e MATRICOLA su ogni foglio consegnato

N.B.: Negli esercizi di programmazione, viene valutata anche la leggibilità del codice proposto. Inoltre, non è consentito l'uso di istruzioni che alterino il normale flusso dell'esecuzione (come, ad esempio, `continue`, `break` e istruzioni di `return` all'interno di cicli che ne provochino l'uscita forzata). Infine non è consentito l'uso di variabili statiche.

Laddove è utilizzato, il tipo `boolean` è definito da `typedef enum {false, true} boolean;`

ESERCIZIO 1 (7 punti)

- Costruire l'NFA che riconosce il linguaggio denotato dall'espressione regolare $(a^* + b^* + (ab)^*)$.
- Ottenere il DFA equivalente, utilizzando la tecnica di costruzione per sottoinsiemi e minimizzare se necessario.
- Scrivere la corrispondente grammatica.

ESERCIZIO 2 (6 punti)

Definire una funzione *ricorsiva* che dato un intero `n`, legge (senza memorizzarla) una sequenza di interi terminata da 0 e calcola la somma degli elementi della sequenza che sono maggiori di `n`. Se ad esempio `n` fosse 5 e la sequenza fosse 13 3 7 2 19 4 0, allora la somma sarebbe 39

ESERCIZIO 3 (7 punti)

Progettare una funzione che dato un array di caratteri di dimensione `dim`, stabilisca se le parentesi eventualmente contenute nell'array sono bilanciate. Ad esempio, se l'array contenesse la seguente sequenza di caratteri: `((a + b)*c)` allora il bilanciamento ci sarebbe, mentre, ad esempio per le sequenze `(a + b + (c)` e per `a +c) + (a` non ci sarebbe.

ESERCIZIO 4 (11 punti)

Sia data una lista (non ordinata) di numeri di telefono, dove ad ogni numero di telefono è associato il numero di chiamate effettuate verso quel numero. Si risolvano i seguenti problemi:

- definire gli opportuni tipi di dato per rappresentare una lista con le specifiche date sopra;
- progettare una procedura *ricorsiva* che, presi in ingresso attraverso opportuni parametri la rappresentazione della lista e un numero di telefono, lo cerca nella lista e, se lo trova, incrementa di uno il numero delle chiamate;
- scrivere una procedura *iterativa* `move2front` che data la lista e un numero di telefono lo cerchi nella lista e sposti l'elemento corrispondente all'inizio della lista, senza modificare l'ordine relativo degli altri elementi.

Negli esercizi precedenti definire in modo chiaro tutte le funzioni o procedure ausiliarie utilizzate per le soluzioni.