

# Fondamenti di Programmazione - CdL in MATEMATICA

## II Prova di verifica del 30/5/2012

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;`

n. eserc.	1	2	3	4
punt. tot	7	7	7	11

### ESERCIZIO 1 (7 punti)

- Sia  $L$  un linguaggio e  $a$  un simbolo, allora si definisce  $L/a$  il *quoziente* di  $L$  e  $a$  come l'insieme delle stringhe  $w$  tali che  $wa \in L$ . Ad esempio se  $L = \{aa, bba, bbb\}$ , allora  $L/a = \{a, bb\}$ . Dimostrare che se  $L$  è regolare anche  $L/a$  lo è.
- Data la seguente grammatica libera

$$\begin{aligned} S &\rightarrow Ab|aaB \\ A &\rightarrow a|Aa \\ B &\rightarrow b \end{aligned}$$

- Dimostrare che è ambigua, mostrando almeno una stringa che si può generare in due modi.
- Descrivere il linguaggio generato e l'NFA che lo riconosce.
- Trovare l'equivalente grammatica regolare.

### ESERCIZIO 2 (7 punti)

Si dice che una sequenza di interi non negativi è *bilanciata* se contiene lo stesso numero di pari e di dispari. Si scriva una funzione *iterativa* che legga una sequenza di interi non negativi (senza memorizzarla) che termina quando vengono immessi consecutivamente due numeri uguali che si suppone facciano entrambi parte della sequenza, restituisca:

- 0 se la sequenza non è ordinata in modo debolmente crescente, né bilanciata;
- 1 se la sequenza non è debolmente crescente, ma è bilanciata;
- 2 se la sequenza è debolmente crescente, ma non è bilanciata;
- 3 la sequenza è debolmente crescente e bilanciata.

### ESERCIZIO 3 (7 punti)

Si definisca una funzione *ricorsiva* che dato un array non vuoto `vet` di caratteri alfabetici minuscoli e la sua dimensione `dim`, restituisca il carattere che precede (rispetto all'ordine alfabetico) tutti i caratteri contenuti nell'array. Se l'array contiene `a` la funzione deve restituire `a`. Se ad esempio l'array contiene la sequenza: `h s l f p`, allora la funzione restituisce il carattere `e`. N.B. Ogni chiamata ricorsiva deve risolvere *esattamente* il problema - posto in origine per tutto l'array - per una porzione dell'array.

### ESERCIZIO 4 (11 punti)

Si vuole modellare il gioco del *domino* mediante una lista concatenata, in cui ogni elemento rappresenta una tessera, in cui sono presenti due sezioni, ciascuna delle quali contiene un numero da 0 a 6.

- Definire i tipi opportuni per rappresentare il gioco.
- Una tessera è inserita in una posizione legale se il suo primo elemento coincide con il secondo elemento della tessera precedente e suo secondo elemento coincide con il primo elemento della tessera successiva. Scrivere una procedura *ricorsiva* che data una disposizione di tessere, elimini la prima tessera che viola l'ordinamento legale.
- Scrivere una funzione *iterativa* che data una disposizione di tessere (non necessariamente legale), e le indicazioni di un'ulteriore tessera, provveda ad inserirla nella prima posizione legale utile, se esiste. La funzione restituisce il valore `true` se questo è possibile, altrimenti restituisce il valore `false`.