

Fondamenti di Programmazione con Laboratorio CdL in MATEMATICA

Appello 10 settembre 2018

N.B.

- Negli esercizi di C, vengono valutati anche l'uso delle condizioni booleane e la leggibilità del codice proposto.
- Non è consentito l'uso di istruzioni che alterino il normale flusso dell'esecuzione all'interno di cicli e che ne provochino l'uscita forzata.
- 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'automa minimo equivalente a quello dato dalla seguente tabella di transizione, dove lo stato iniziale è q_0 , mentre q_3 and q_5 sono stati finali:

		0	1
→	q_0	q_1	q_3
	q_1	q_0	q_2
	q_2	q_5	q_4
*	q_3	q_5	q_3
	q_4	q_4	q_7
*	q_5	q_5	q_4
	q_6	q_4	q_1
	q_7	q_4	q_7

- Scrivere la corrispondente grammatica regolare.

ESERCIZIO 2 (9 punti)

Dire se i seguenti linguaggi sono regolari o liberi dal contesto, giustificando la risposta:

- $L_1 = \{a^n b^{n+1} \mid n > 0\}$
- $L_2 = \{(abc)^n \mid n \geq 1\}$
- $L_3 = \{a^n b^n c^{n+1} \mid n \geq 1\}$

ESERCIZIO 3 (7 punti)

Supponiamo di voler realizzare un metodo di cifratura di una frase che consiste nel riscrivere la frase **al contrario** utilizzando la regola che prevede che ogni carattere sia rimpiazzato dal suo corrispondente secondo una codifica predefinita.

Supponendo di disporre già della funzione `code` che fornisce la codifica del carattere puntato dal suo argomento, con il prototipo `char code (char)`, si scriva una funzione **ricorsiva** e **senza utilizzare costrutti iterativi** che

- legga da standard input una frase terminata da `'\n'` e
- stampi in ordine inverso la cifratura della stessa,
- restituendo il numero di caratteri.

Ad esempio, supponendo che la codifica di un carattere minuscolo (risp. maiuscolo) sia il corrispondente carattere maiuscolo (risp. minuscolo) e che la codifica carattere spazio (rappresentato con `~`) sia il carattere `#`, la cifratura della frase

`la~mia~cAsa`

è la seguente

`ASaC#AIM#AL`

mentre il numero di caratteri restituito è: 11.

ESERCIZIO 4 (7 punti)

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