

Fondamenti di Programmazione - CdL in MATEMATICA

II Prova di verifica del 30/5/2013

num. eserc.	1	2	3	4
punt. tot	7	7	6	10

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).
- 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)

- Mostrare che il linguaggio L delle stringhe su $\{0,1\}$ tali che il numero di occorrenze di 01 è uguale al numero di occorrenze di 10 è *regolare*. N.B.: 010 (e anche 1 e 00100) appartengono ad L mentre 0101 no.
- Minimizzare l'automa, se necessario, altrimenti mostrare che è già minimo.
- Fornire la grammatica regolare che genera L .

ESERCIZIO 2 (7 punti)

Si consideri un array di dimensione `dim`, chiamato `formula` di caratteri 'F' (False `ff`), 'T' (True `tt`), 'A' (And \wedge), 'O' (Or \vee) che rappresenta una formula booleana *ben formata*, con la convenzione che gli operatori 'And' e 'Or' abbiano la stessa priorità e associno a destra. In altre parole si tratta di formule generabili a partire dalla seguente grammatica

$$\begin{aligned} E &\rightarrow T \wedge E | T \vee E | T \\ T &\rightarrow \text{ff} | \text{tt} \end{aligned}$$

Scrivere la funzione **ricorsiva** `check-ric` che, dato un array di caratteri, valuti la formula booleana data. Ad esempio se `formula` fosse il seguente array

'T'	'O'	'F'	'A'	'T'	'O'	'F'
-----	-----	-----	-----	-----	-----	-----

allora la corrispondente formula booleana sarebbe $(\text{tt} \vee (\text{ff} \wedge (\text{tt} \vee \text{ff})))$ (vedi la derivazione corrispondente) e la funzione `check-ric` dovrebbe restituire `true`.

ESERCIZIO 3 (6 punti)

Scrivere una funzione **iterativa** che, dato un array `a` di interi, di dimensione `dim` con `dim` multiplo di 3, controlli che sia composto da una serie di *terne pitagoriche* (terne di numeri naturali a, b, c tali che $a^2 + b^2 = c^2$) consecutive a partire dalla prima posizione (le terne sono disgiunte e non si sovrappongono, quindi la prima si trova alle posizioni 0,1,2, la seconda alle posizioni 3,4,5, e così via). La funzione deve restituire `true` se sono tutte terne pitagoriche, `false` altrimenti. La funzione deve anche stampare il numero di terne pitagoriche presenti nell'array dopo l'eventuale prima terna non pitagorica.

Ad esempio se l'array è

3	4	5	11	60	61	28	45	53
---	---	---	----	----	----	----	----	----

la funzione restituisce `true` e stampa 0, se invece è

3	4	5	11	60	50	28	45	53
---	---	---	----	----	----	----	----	----

la funzione restituisce `false` e stampa 1.

ESERCIZIO 4 (10 punti)

Si vuole modellare una lista che rappresenta il *magazzino* di una libreria. Ogni elemento della lista rappresenta un libro e deve contenere le seguenti informazioni:

- un campo numerico che identifica l'autore;
- un campo numerico che indica il titolo;

- un campo numerico che indica il numero di copie.

Per risolvere l'esercizio si facciano le seguenti cose.

- Definire i tipi opportuni per rappresentare una siffatta lista.
- Supponendo che la lista sia *ordinata* in base al campo numerico degli autori, scrivere una funzione *ricorsiva* che, data la lista di libri, il codice autore, e il codice titolo, cerchi l'elemento corrispondente e diminuisca di **uno** il numero di copie, eliminando l'elemento corrispondente se il numero scende a **zero**. La funzione restituisce **true** se è stata trovata una copia del libro cercato, **false** altrimenti.
- Supponendo che la lista sia *ordinata* in base al campo numerico degli autori, scrivere una procedura che, dati i codici numerici relativi all'autore, al titolo e al numero delle copie, inserisca nella lista un nuovo elemento (che si suppone non esista ancora nella lista) popolato con questi dati, che rispetti l'ordine della lista.