

Fondamenti di Programmazione - CdL in MATEMATICA

Appello del 11/1/2014

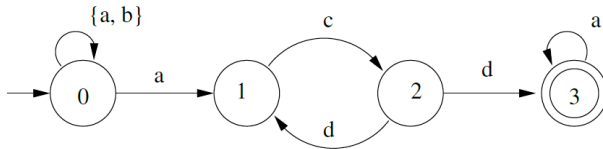
num. eserc.	1	2	3	4
punt. tot	7	6	6	11

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)

Dato l'automa non deterministico A descritto dalla seguente tabella di transizione



- Descrivere il linguaggio $L(A)$, accettato da A
- Disegnare l'automa deterministico equivalente, usando la costruzione per sottoinsiemi.
- Scrivere la grammatica *regolare* equivalente.

ESERCIZIO 2 (6 punti)

Scrivere una funzione *ricorsiva* che, legga una sequenza di interi da input, terminata da 0 e che restituisca `true` se il numero di elementi negativi supera il numero di elementi positivi di al più n unità, e che restituisca `false` in caso contrario. Ad esempio, se la sequenza fosse

2, -7, -5, -4, 6, -4, 5, 0

ed n fosse 2, la funzione dovrebbe restituire `true`. Si risolva il problema, senza utilizzare (i) costrutti iterativi, né (ii) strutture di appoggio, tipo liste o vettori.

ESERCIZIO 3 (6 punti)

Si scriva in C una funzione che ha in input gli array `a`, `b` e `c`, di dimensione `dimA`, `dimB` e `dimC = dimA + dimB` di interi positivi distinti. Si scriva una funzione `C unione` che modifichi l'array `c` in modo che contenga tutti gli elementi di `a` seguiti dagli elementi di `b` che non sono presenti in `a`, e restituisca l'indice in `c` dell'ultimo elemento inserito. Ad esempio, sugli array

12	4	8	1	5				
30	1	45	12					
-1	-1	-1	-1	-1	-1	-1	-1	-1

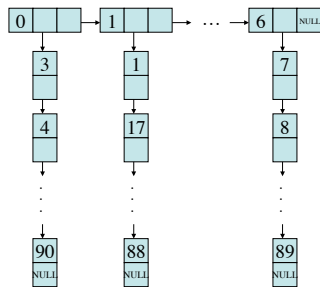
la funzione modifica l'array `c` come di seguito e restituisce il valore 6, essendo 45 l'ultimo elemento inserito.

12	4	8	1	5	30	45	-1	-1
----	---	---	---	---	----	----	----	----

ESERCIZIO 4 (11 punti)

Si vuole modellare una lista di cartelle della *Tombola*. In ogni cartella, ogni elemento contiene il campo per il numero, e gli elementi sono ordinati in modo crescente.

Vogliamo quindi avere una lista concatenata, dove ogni elemento rappresenta una cartella, ovvero una sottolista, come illustrato nella figura sottostante.



- (1 punti) Definire i tipi opportuni per la rappresentazione indicata.
- (5 punti) Scrivere una funzione *iterativa* che dato un numero n cerchi ed elimini (laddove esiste), da ogni cartella, l'elemento che contiene n .
- (3+2 punti) Scrivere una funzione *ricorsiva* che, dato un intero m e data la lista delle cartelle, individui ed elimini l' m -sima cartella, i cui elementi siano stati tutti eliminati. Si premieranno le soluzioni in cui non si aggiungano altri parametri oltre alla testa della lista e al numero m .