

Fondamenti di Programmazione - CdL in MATEMATICA

II Prova di verifica del 1/6/2011

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;**

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

ESERCIZIO 1 (7 punti)

Si dica, argomentandolo, se il linguaggio L composto da stringhe su $\Sigma = \{a, b\}$ tali che:

- ci sono almeno due a consecutive e
- non ci sono mai due b consecutive

è un linguaggio regolare. Se lo è, si determini l'automa minimo che riconosce L .

ESERCIZIO 2 (7 punti)

Scrivere una procedura *ricorsiva* che legga (senza memorizzarla) una sequenza di interi non negativi $a_0, a_1, \dots, a_i, \dots$, terminata da -1 e che stampi, in ordine inverso, solo gli a_i che sono preceduti da a_{i-1} tali che $a_{i-1} < a_i$, escludendo -1 . L'ultimo elemento da stampare il primo della sequenza. Se per esempio la sequenza in ingresso fosse: 3 2 8 7 -1 la stampa dovrebbe essere 8 3.

ESERCIZIO 3 (7 punti)

Si definisca una funzione

boolean controlla (int v[], int dim)

che restituisce **true** se vale la seguente proprietà:

$$\exists i \in [0, dim - 2]. (pari(v[i]) \vee (\forall j \in [i + 1, dim - 1]. v[j] < v[i]))$$

ESERCIZIO 4 (11 punti)

Si vuole modellare un calendario accademico, ovvero una sequenza ordinata di date consecutive, mediante una lista concatenata, in cui ogni elemento rappresenta un'offerta con:

- un campo relativo alla data, contenente giorno e mese;
- un campo che riporta se nella data è possibile fissare un esame o no;
- un campo che riporta se in quella data è stato fissato un esame oppure no.
- Definire i tipi opportuni per rappresentare un siffatto calendario.
- Scrivere una procedura *iterativa* che dato un calendario, trovi due date a distanza di almeno 20 giorni l'una dall'altra in cui fissare due appelli, cambiando quindi l'apposito campo nei due elementi corrispondenti. Si suppone che trovare due date con queste caratteristiche sia possibile.
- Scrivere una funzione *ricorsiva* che dato un calendario dove il primo appello si è appena concluso, provveda ad eliminare tutte le date fino a quella del primo appello incluso.