

FONDAMENTI DI PROGRAMMAZIONE - CdL in MATEMATICA

PROVA SCRITTA DEL 16/06/2010

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;`

ESERCIZIO 1 (5 punti)

Dato il seguente automa a stati finiti non deterministico:

		$\{a\}$	$\{b\}$
\rightarrow	q_0	$\{q_0, q_2\}$	$\{q_0, q_1, q_2\}$
	q_1	\emptyset	$\{q_1\}$
\star	q_2	$\{q_2\}$	$\{q_2\}$

- (i) trasformarlo in un automa deterministico equivalente, utilizzando la tecnica della costruzione dei sottinsiemi;
- (ii) indicare formalmente il linguaggio riconosciuto dall'automata;
- (iii) definire una grammatica **regolare** che generi lo stesso linguaggio.

ESERCIZIO 2 (7 punti)

Supponiamo di voler realizzare un metodo di cifratura di una frase che consiste nel riscrivere la frase al contrario utilizzando le seguenti regole

- ogni carattere è rimpiazzato dal suo corrispondente secondo una codifica predefinita
- le vocali presenti nella frase vengono triplicate

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

`la~mia~cAa`

è la seguente

`AAASaaaC#AAAIIM#AAAL`

Supponendo di disporre della funzione `code` col seguente prototipo

`char code (char)`

che fornisce la codifica del carattere puntato dal suo argomento, si scriva, **senza utilizzare costrutti iterativi**, una funzione che legge da standard input una frase terminata da `'\n'` e stampa la cifratura della stessa, e restituisce il numero di consonanti.

ESERCIZIO 3 (7 punti)

Dati due array `a` e `b` di dimensione `dim` in cui gli elementi di `b` sono interi distinti nell'intervallo `[0, dim-1]`, diciamo che `a` è **b-ordinato** se gli elementi di `a`, presi nell'ordine specificato da `b`, sono ordinati in senso crescente. Ad esempio, dato l'array `a` rappresentato in figura

10	5	30	2	20
----	---	----	---	----

se l'array `b` è

3	1	0	4	2
---	---	---	---	---

allora `a` è **b-ordinato**, mentre se l'array `b` è

2	1	4	0	3
---	---	---	---	---

allora `a` **non** è **b-ordinato**.

Scrivere una funzione `C` con prototipo

`boolean check(int vet1[], int vet2[], int dim)`

che restituisce **true** se `vet2` è **vet1-ordinato**, restituisce **false** altrimenti.

ESERCIZIO 4 (11 punti)

Una macchinetta per la distribuzione di gelati viene gestita mediante una lista collegata in cui ciascun elemento della lista contiene:

- il codice numerico del gelato
- il numero di confezioni del gelato ancora disponibili nella macchinetta (un valore intero strettamente positivo)

La lista è ordinata per numero crescente di codice. Dato un gelato con un codice c , la macchinetta può contenere al più un elemento con quel codice. Si risolvano i seguenti problemi

- definire gli opportuni tipi di dato per rappresentare una macchina con le specifiche date sopra
- progettare una procedura **ricorsiva** che, presi in ingresso attraverso opportuni parametri la rappresentazione di una macchinetta e il codice di un gelato, se questo è disponibile diminuisce di uno il numero di confezioni presenti, e lo elimina dalla macchinetta qualora il numero di confezioni disponibili diventi nullo. La procedura deve anche stampare su standard output l'esito dell'operazione ("**Ritirare il prodotto**" oppure "**Prodotto non disponibile**")
- progettare una procedura **iterativa** che, presi in ingresso attraverso opportuni parametri una macchinetta, un nuovo codice e un numero i , inserisca nella macchinetta i gelati con quel codice.

Negli esercizi precedenti definire in modo chiaro tutte le funzioni o procedure ausiliarie utilizzate per le soluzioni.