

# Fondamenti di Programmazione - CdL in MATEMATICA

## I Prova di verifica del 15/4/2014

num. eserc.	1	2	3	4	5
punt. tot	10	7	9	2	2

**N.B.:** Negli esercizi di programmazione, vengono valutati anche l'uso delle condizioni booleane e la leggibilità del codice proposto. Inoltre, non è consentito l'uso di istruzioni che alterino il normale flusso dell'esecuzione all'interno di cicli e provochino l'uscita forzata. Infine, è possibile presupporre le seguenti direttive di compilazione:

```
# define FALSE 0
# define TRUE 1
```

### ESERCIZIO 1 (10 punti)

Sia  $L$  il linguaggio regolare, definito su  $\Sigma = \{a, b, c\}$ , come segue

$$L = \{a^n b^m c^k | n \geq 0, m > 0, k \geq 0\}$$

- Fornire l'automa a stati finiti non deterministico (NFA)  $N$  che riconosce  $L$ .
- Costruire l'automa a stati finiti deterministico (DFA) equivalente  $D$ , usando la costruzione per sottoinsiemi.
- Ricavare a partire da  $D$ , l'espressione regolare  $R$ , tale che  $L(R) = L(D)$ , utilizzando la tecnica di eliminazione di stati e trascurando gli eventuali stati pozzo.
- Semplificare, se possibile, l'espressione ottenuta, ricorrendo alle leggi algebriche delle espressioni regolari.

### ESERCIZIO 2 (7 punti)

Sia  $L$  il linguaggio regolare, definito come nell'Esercizio 1.

Scrivere in C una funzione `riconosce` che, dato un array `v` di caratteri e la sua dimensione `dim`, restituisce `TRUE` se la sequenza `v[0] v[1] ... v[dim-1]` appartiene al linguaggio  $L$ . Ad esempio, la funzione darebbe `TRUE` se applicata a

a	a	a	b	b	b	b	c	c	c	c
---	---	---	---	---	---	---	---	---	---	---

### ESERCIZIO 3 (9 punti)

- Definiamo una sequenza *tramezzata* come una sequenza di interi, in cui si ripete costantemente la prima coppia di numeri  $ij$  immessa (con  $i \neq j$ ) (ad esempio  $3\ 6\ 3\ 6\ 3\ 6\ 3\ 6\ 3\ 6\ \dots$ ). La sequenza termina non appena si interrompe la "tramezzatura" e quindi la sua lunghezza non è nota prima dell'esecuzione. Si definisca una funzione C che, riceve in ingresso una sequenza tramezzata di interi di lunghezza arbitraria, ma lunga almeno 2, e che controlli se il numero di coppie immesse è pari. Ad esempio se la sequenza data fosse  $10\ 3\ 10\ 3\ 10\ 3\ 8$  la funzione dovrebbe restituire `FALSE`. Per quanto riguarda il `main`, occorre solo scrivere la chiamata della funzione.
- Dati in ingresso due interi  $i$  e  $j$ , definiamo una sequenza *pseudo-tramezzata* come una sequenza di interi, in cui ogni occorrenza dell'intero  $i$  deve essere immediatamente seguita da almeno un'occorrenza intero  $j$ . La sequenza termina non appena questa condizione viene violata. Si modifichi la funzione del punto precedente, in modo da controllare se il numero di coppie  $ij$  è pari. Ad esempio, se  $i$  e  $j$  fossero 4 e 7 e la sequenza fosse  $6\ 9\ 4\ 7\ 5\ 9\ 3\ 7\ 4\ 7\ 4\ 5$ , la funzione dovrebbe restituire `true`.

### ESERCIZIO 4 (2 punti)

Completare la specifica dello *stato finale*, relativa al seguente problema. Data una sequenza di interi di dimensione  $dim$ , e un valore  $val$ , calcolare il numero di elementi dispari *occDisp* e controllare se questo coincide con  $val$ .

Stato iniziale:  $\{\dim \rightsquigarrow K, c[0] \rightsquigarrow V_0, \dots, c[K-1] \rightsquigarrow V_{K-1}, val \rightsquigarrow V\}$  con  $K > 0$   
Stato finale:  $\{\dots\}$

N.B. Non occorre scrivere il programma.

(SEGUE ALTRA PAGINA)

### ESERCIZIO 5 (2 punti)

Data la funzione `misteriosa`

```
main()
{
float *q, y;
char c = 'a';
q = &y;
c = misteriosa(&c,&q);
...
}
```

dire qual è deve essere il tipo dei parametri formali della funzione, perché la chiamata sia corretta.