

# Fondamenti di Programmazione - CdL in MATEMATICA

## II Appello del 12/2/2015

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)

Dati l'alfabeto  $\Sigma = \{0, 1\}$  sia  $L$  il linguaggio

$$L = \{w_1 \dots w_n \mid n > 0 \wedge w_i \in L_1, i \in [1, n]\}$$

dove  $L_1 = \{011, 0111\}$ .

- Si definisca un automa a stati finiti non deterministico che riconosce  $L$ .
- Utilizzando la tecnica di *costruzione per sottoinsiemi* si ricavi il corrispondente automa a stati finiti deterministico.
- Si fornisca la corrispondente grammatica *regolare*.

### ESERCIZIO 2 (7 punti)

Si consideri la seguente grammatica  $G = \langle \{a, b, c\}, \{S, A\}, S, P \rangle$ , in cui le produzioni in  $P$  sono:

$$\begin{aligned} S & ::= aSc|A \\ A & ::= bAc|\epsilon \end{aligned}$$

- Scrivere qual è il linguaggio  $L$  generato da  $G$ .
- Scrivere in C una o più funzioni *ricorsive* che dato un array `a` di caratteri alfabetici minuscoli e la sua dimensione `dim` controllino se la sequenza contenuta nell'array appartiene o meno al linguaggio generato dalla grammatica  $G$ .

### ESERCIZIO 3 (6 punti)

Si scriva una funzione C `boolean controlla (int v[], int dim)` che, dato un array `v` e la sua dimensione `dim`, restituisca il valore di verità della seguente formula

$$\forall i \in [0, dim - 1). \exists j \in [i + 1, dim) : v[i] < v[j]$$

### ESERCIZIO 4 (10 punti)

Data una lista di interi definita come:

```
struct el {int info; struct el *next;};
typedef struct el ElementoDiLista;
typedef ElementoDiLista *ListaDiElementi;
```

- Scrivere una funzione *iterativa* C che, date due liste di interi *ordinate* in ordine crescente, inserisca nella prima lista, rispettando l'ordine, tutti i valori della seconda lista non presenti nella prima e ne restituisca il numero.
- Scrivere in C una procedura *ricorsiva* che, data una lista di interi, elimini dalla lista tutti gli elementi in posizione pari.