

Fondamenti di Programmazione con Laboratorio CdL in MATEMATICA

Prima Verifica Intermedia 12 novembre 2018

N.B.: Negli esercizi di programmazione, vengono valutati anche l'uso delle condizioni booleane, la leggibilità e l'efficienza del codice proposto. Inoltre, non è consentito l'uso di variabili globali e 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 (circa 9 punti)

- Costruire l'NFA A per il linguaggio L composto da tutte le stringhe w in $\{0, 1, 2\}^*$ che **contengano** 01 o 11.
- Ricavare il DFA corrispondente ad A , **usando** l'algoritmo di costruzione per sottoinsiemi.
- Costruire il DFA A' che riconosce il linguaggio L' composto da tutte le stringhe w in $\{0, 1, 2\}^*$ tali che **non contengano** occorrenze né di 01, né di 11.

ESERCIZIO 2 (circa 7 punti)

Si scriva in C una funzione (non occorre scrivere il `main`) che

- legga da linea di comando (senza usare array di appoggio) una sequenza di interi positivi n_1, n_2, \dots che termina non appena l' i -esimo numero letto della sequenza n_i risulti essere il successivo dell' $(i-1)$ -esimo numero n_{i-1} ;
- restituisca l'indice i corrispondente. Ad esempio la funzione restituirebbe 11 se la sequenza letta fosse:

6, 8, 49, 34, 99, 32, 98, 5, 3, 77, 78

- dire come cambierebbe la condizione di terminazione della lettura nel caso si leggano caratteri e si debba controllare che il carattere appena letto sia una lettera maiuscola e che al tempo stesso sia la lettera successiva (rispetto all'ordine alfabetico) a quella letta immediatamente prima.

ESERCIZIO 3 (circa 8 punti)

Si scriva in C una funzione (non occorre scrivere il `main`) che, dato un array a di interi, di dimensione dim ,

- controlli che ogni elemento $a[i]$, di indice dispari i , sia il quadrato dell'elemento precedente.
- La funzione deve restituire `TRUE` se la proprietà è rispettata, `FALSE` altrimenti.

La funzione restituirebbe `FALSE` se ad esempio l'array fosse

3	9	5	25	9	81	28	45
---	---	---	----	---	----	----	----

ESERCIZIO 4 (circa 3 punti)

Si descriva a parole (massimo 10 righe) l'algoritmo **INSERTION SORT**, e se ne scriva l'*invariante di ciclo*, ovvero la proprietà sullo stato che è vera all'inizio o alla fine di ogni iterazione.

ESERCIZIO 5 (circa 3 punti)

• Date due espressioni booleane b_0, b_1 e uno stato σ , scrivere la valutazione dell'espressione composta $b_0 \wedge b_1$, in cui \wedge rappresenta l'AND, ovvero completare la regola premessa/conclusione della semantica operativa, dove t è un valore booleano:

$$\frac{\dots \quad \dots}{\langle b_0 \wedge b_1, \sigma \rangle \rightarrow t}$$

- come cambierebbe la regola con la valutazione così detta *lazy*?

ESERCIZIO 6 (circa 2 punti)

```
int x, y, *p, *q;
x=6;
y=12;
p = &x;
q = &y;
*q = x + y;
x = x + *q;
q = p;
*q = x + y;
```

Mostrando i passaggi, dire quali sono alla fine i valori legati ad x e y .