

# Fondamenti di Programmazione - CdL in MATEMATICA

## I Prova di verifica del 15/11/2017

**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 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 (12 punti)

- Dato il seguente automa a stati finiti non deterministico (NFA)  $A$  sull'alfabeto  $\Sigma = \{a, b, c, d\}$ :

		$a$	$b$	$c$	$d$
$I$	$q_0$	$\{q_0, q_1\}$	$\{q_0\}$	$\emptyset$	$\emptyset$
	$q_1$	$\emptyset$	$\emptyset$	$\{q_2\}$	$\emptyset$
	$q_2$	$\emptyset$	$\emptyset$	$\emptyset$	$\{q_1, q_3\}$
$F$	$q_3$	$\{q_3\}$	$\emptyset$	$\emptyset$	$\emptyset$

- fornire la corrispondente espressione regolare, utilizzando l'algoritmo di eliminazione degli stati, applicandolo (in questo caso si può) direttamente all'NFA.
  - fornire l'automa a stati finiti deterministico (DFA) equivalente risultante dall'applicazione dell'algoritmo di costruzione dei sottoinsiemi;
- Sia  $L$  il linguaggio su  $\Sigma = \{a, b\}$  formato da tutte le stringhe che contengono almeno una  $a$  nelle ultime due posizioni. Ad esempio  $aabbbab \in L$ .
    - Descrivere un  $\epsilon$ -NFA (che usa  $\epsilon$ -transizioni) che riconosca  $L$ .
    - Scrivere le  $\epsilon$ -chiusure degli stati da cui escono  $\epsilon$ -transizioni.

### ESERCIZIO 2 (8 punti)

Completare il seguente programma in C (**N.B.** senza modificare la parte di codice data) in modo che contenga ed usi una funzione in grado di:

- ricevere due array  $a$  e  $b$  entrambi di dimensione  $DIM$  di interi;
- verificare le seguenti due proprietà:
  1. ogni elemento contenuto nell'array  $a$  in una posizione con *indice* pari è presente anche nell'array  $b$  (in una posizione qualunque);
  2. ogni elemento contenuto nell'array  $a$  in una posizione con *indice* dispari è minore di tutti gli elementi contenuti in  $b$ ;

La funzione deve restituire 1 se la proprietà è verificata, 0 altrimenti.

Dato per esempio  $DIM$  uguale a 6 e i seguenti array, la funzione restituirà 1.

34	25	44	17	40	11	56	44	34	46	26	40
----	----	----	----	----	----	----	----	----	----	----	----

Si possono usare funzioni ausiliarie, basta che le si definisca.

```
#include <stdio.h>
#define DIM ...
```

```
/* DICHIARAZIONE E DEFINIZIONE DELLA FUNZIONE RICHIESTA: DA COMPLETARE */
```

```
main() {
    int a[DIM], b[DIM];
    int i, check-res;
    for (i=0; i<DIM; i++)
        {scanf("%d", &a[i]);}
    for (i=0; i<DIM; i++)
        {scanf("%d", &b[i]);}
    check-res = Controlla(-,-,-); <----- CHIAMATA DELLA FUNZIONE: DA COMPLETARE
    printf("Il risultato e' %d\n", check-res);
}
```

### ESERCIZIO 3 (6 punti)

Scrivere una funzione che,

- legga una sequenza di caratteri alfabetici da input (senza memorizzarla) che termina quando viene immesso il carattere `\n` e che
- restituisca 1 se il numero di caratteri maiuscoli supera il numero di caratteri minuscoli di al più `n` unità, e che restituisca 0 in caso contrario.

Ad esempio, se `n` fosse 2, e la sequenza fosse

`A, b, k, G, R, U, f, \n`

la funzione dovrebbe restituire 1, essendo 4 i caratteri maiuscoli e 3 quelli minuscoli.

**N.B.** Si risolva il problema senza utilizzare strutture di appoggio, tipo array.

### ESERCIZIO 4 (2 punti)

Sia data la procedura `C` e il `main` riportati di seguito.

```
void Boh(int x,int *y)
{
x = *y;
*y = *(y+1);
*(y+1) = x;
}

int main()
{
int a[] = {11,22,33,44};
Boh(a[0],&a[1]); \* (PUNTO 1) *\
return 0;
}
```

Scrivere cosa fa la procedura e come cambia il contenuto dell'array `a` dopo la chiamata (PUNTO 1) della funzione nel corpo del `main`.

### ESERCIZIO 5 (2 punti)

Completare la specifica dello *stato finale*, relativa al seguente problema. Data una sequenza di interi controllare, usando una variabile *check*, se il numero di elementi pari è uguale al numero di elementi dispari.

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

**N.B.** **Non** si deve scrivere il programma.