

Fondamenti di Programmazione - CdL in MATEMATICA

I Prova di verifica del 13/4/2010

| | | | | |
|-------------|---|---|---|---|
| num. eserc. | 1 | 2 | 3 | 4 |
| punt. tot | 8 | 6 | 8 | 8 |

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 all'interno di cicli che ne provochino l'uscita forzata.

ESERCIZIO 1 ((8) punti)

- (4 punti) Definire sull'alfabeto $\Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ un automa a stati finiti **deterministici** (DFA) D che riconosca il linguaggio dato dalle sole stringhe di cifre decimali che rappresentano numeri divisibili per 3:

$$L = \{c_n \dots c_0 \mid n \geq 0, c_i \in \Sigma, (\sum_{i=0}^n c_i * 10^i) \bmod 3 = 0\} \cup \{\epsilon\}$$

spiegando il principio seguito nella costruzione dell'automa.

- (4 punti) Si definisca l'automa analogo D' definito sull'alfabeto $\Sigma' = \{0, 1, 2\}$, cioè quello che riconosce:

$$L = \{t_n \dots t_0 \mid n \geq 0, t_i \in \Sigma', (\sum_{i=0}^n t_i * 10^i) \bmod 3 = 0\} \cup \{\epsilon\}$$

e si dimostri formalmente che D' riconosce il linguaggio richiesto.

ESERCIZIO 2 (6 punti)

Dato il seguente programma:

```
#include <stdio.h>
main()
{
    int x, y, z;
    scanf("%d %d %d ", &x, &y, &z);
    while ((x != y) || (y != z))
    {
        if (x != y)
            y = y - 1;
        else
            if (y != z)
                z = z + 1;
    }
    printf("%d %d %d \n", x, y, z);
}
```

Individuare i valori iniziali x_0, y_0, z_0 delle variabili x, y, z , per i quali:

- il programma termina e produce in output la stessa terna fornita in input;
- il programma termina e produce in output una terna diversa da quella fornita in input;
- il programma non produce alcun output, ovvero non termina.

ESERCIZIO 3 (8 punti)

Sia x_1, \dots, x_n, \dots una sequenza di interi *positivi*. Diciamo che l'indice k *interrompe la sequenza*, ovvero che la sequenza non legge nient'altro dopo x_k , se k è il minimo naturale tale che:

$$\sum_{i=1}^k x_i \geq 10 * \prod_{i=1}^k i \quad (1)$$

Scrivere un programma che legga una sequenza di interi **positivi** fino all'indice k e su questa sequenza determini il

- Il numero *mul* di occorrenze di multipli di 3;
- La posizione *pos* della seconda occorrenza, se esiste, di multipli di 3.

Attenzione, k non è noto a programma, ma va determinato a tempo di esecuzione, controllando la condizione (1).

ESERCIZIO 5 ((6+2) punti)

Completare il seguente programma in modo che controlli che:

- gli elementi pari appaiano in ordine crescente
- gli elementi dispari appaiano in ordine decrescente

Dato ad esempio il vettore

| | | | | | | |
|----|----|----|----|----|---|----|
| 12 | 49 | 23 | 16 | 11 | 5 | 18 |
|----|----|----|----|----|---|----|

si ottiene la stampa seguente: Il controllo ha dato esito positivo

Dato invece il vettore

| | | | | | | |
|----|----|----|----|----|---|---|
| 49 | 12 | 23 | 16 | 11 | 5 | 8 |
|----|----|----|----|----|---|---|

si ottiene la stampa seguente: Il controllo ha dato esito negativo

Attenzione:

- I due punti in più si danno nel caso in cui tutti i controlli vengano fatti all'interno di un solo ciclo.
- Non è consentito modificare la parte di codice data.
- Per semplicità si ipotizza che tra i primi due numeri dell'array, uno sia pari e uno sia dispari.

```
#define DIM ...
/* Si suppone che DIM >= 2 */
main()
{
    int vet[DIM];
    int i; int trovato = 0;

    /* AGGIUNGERE QUI EVENTUALI ALTRE DICHIARAZIONI DI VARIABILE */

    for (i=0; i<DIM; i++)
        scanf("%d", &vet[i]);

    /* CODICE DA COMPLETARE */

    if (trovato == 0)
        printf("Il controllo ha dato esito positivo" );
    else
        printf("Il controllo ha dato esito negativo" );
}
```