

Fondamenti di Programmazione - CdL in MATEMATICA

I Prova di verifica del 14/11/2014

num. eserc.	1	2	3	4	5
punt. tot	12	8	6	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 (12 punti)

- Dato il seguente linguaggio sull'alfabeto $\Sigma = \{1, 2, 3\}$:

$$L = \{x13y \mid x \in \Sigma^*, y \in \{1, 2\}^*\}$$

- Definire un automa a stati finiti non deterministico (NFA) che riconosca L , e
- trasformarlo in un automa deterministico (DFA) equivalente, utilizzando la tecnica della costruzione dei sottoinsiemi.
- – Dato l'alfabeto $\Sigma = \{1, 2, 3\}$, definiamo come stringa *non decrescente* una stringa, dove un 2 non può apparire prima di un 1 e dove un 3 non può apparire prima di un 2 o di un 1. Ad esempio, 1, 222, 2, 1133, 2223 sono stringhe non decrescenti, mentre 22221 e 321 no. Definire un ϵ -NFA che riconosca le stringhe del seguente linguaggio sull'alfabeto Σ :

$$L' = \{x \mid x \in \Sigma^*, x \text{ non decrescente}\}$$

- Definire un NFA o un DFA che riconosca le stringhe di L' .

ESERCIZIO 2 (8 punti)

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

- ricevere un array di dimensione DIM di elementi compresi nell'intervallo tra 0 e 99 e un numero $N > 0$;
- identificare le dieci decine in cui si può suddividere l'intervallo tra 0 e 99: con il numero 0 per la decina che va da 0 a 9, con il numero 1 per quella che va da 10 a 19, ..., con il numero 9 quella che va da 90 a 99;
- e restituire il numero corrispondente a una delle decine che hanno **almeno** N rappresentanti nel vettore dato. Restituirà -1 se tale decina non c'è.

Dato per esempio N uguale a 3 e il seguente vettore, la funzione restituirà 4, dato che la decina richiesta è quella tra 40 e 49.

4	7	48	17	21	41	34	49	87	46	26	53
---	---	----	----	----	----	----	----	----	----	----	----

```
#define DIM ...
#define N ...
```

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

```
main() {
int vet[DIM]; /* assumo gli interi tutti compresi tra 0 e 99 */
int i, req_dec;
for (i=0; i<DIM; i++)
    {scanf("%d", &vet[i]);}
req_dec = ControllaDecine(vet, DIM, N);
printf("La decina richiesta e' %d\n", req_dec);
}
```

ESERCIZIO 3 (6 punti)

Scrivere una procedura void `Cifrario(int k)` (non occorre scrivere il `main()`) che, dato un intero `k`:

- legga da linea di comando una sequenza di caratteri che termina quando viene immesso un carattere (che si suppone **non** faccia parte della sequenza) uguale al primo carattere letto della sequenza;
- ristampi i caratteri, mano a mano che vengono letti, rimpiazzando ogni singolo carattere alfabetico con un nuovo carattere posto `k` (modulo 26) posizioni avanti nell'alfabeto (Gli altri caratteri vengono ristampati senza modifiche.)

Ad esempio, se $K = 3$ allora la sequenza seguente

```
*Attaccare i galli...*
```

verrebbe ristampata come:

```
*Dwwdffduh n ldoon...
```

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 massimo e controllare se questo coincide con *val*.

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

Stato finale: $\{ \text{massimo} \rightsquigarrow \dots, \text{check} \rightsquigarrow \dots \}$

Attenzione. Non si deve scrivere il programma.

ESERCIZIO 5 (2 punti)

Riscrivere il seguente frammento di codice senza utilizzare il comando `switch`.

```
switch (y)
{
'a': x=x+1;
'b': x=x+2; break;
'c': x=x+3;
}
```