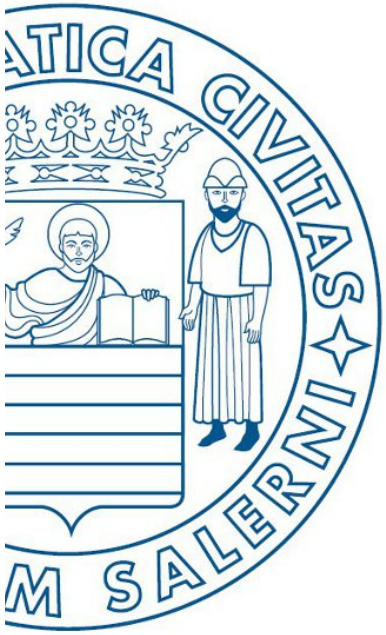


Università di Salerno
Dipartimento di
Ingegneria Industriale

UNIVERSITÀ DEGLI STUDI DI SALERNO

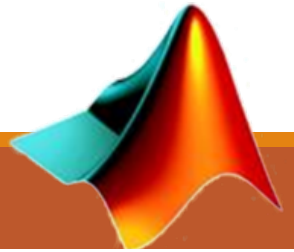
Fondamenti di Informatica

Esercizi Ripasso Argomenti MATLAB – PARTE I - Soluzioni

Prof. Christian Esposito

Corso di Laurea in Ingegneria Meccanica e Gestionale (Classe I)

A.A. 2016/17



MATLAB

N

<<noleggio.txt>>	1	2	3	4	5	6	7	8
Auto 1 (indice 1)	**1**	**1**	**1**	**0**	**0**	**1**	**1**	**0**
Auto 2 (indice 2)	**0**	**0**	**1**	**1**	**0**	**1**	**0**	**0**
Auto 3 (indice 3)	**1**	**1**	**0**	**0**	**0**	**0**	**1**	**1**

ore

I

<<incassi.txt>>	Auto 1	Auto 2	Auto 3
Incasso Orario	**8**	**10**	**5**

- In questa esercitazione verranno utilizzate una matrice **N** ed un array riga **I**
  - La matrice **N** rappresenta **il prospetto giornaliero** di un autonoleggio
  - L'elemento **`N(riga, colonna)`** assume valore `1` se l'auto specificata dalla `riga` è noleggiata all'ora specificata dalla `colonna`, valore `0` altrimenti
    - ***Esempio***: `N(2, 7)` → ha valore `0` ed indica che l'*Auto 2* (riga 2) è *libera* (*non noleggiata*) all'ora `7` (colonna 7)
    - ***Esempio***: `N(3, 2)` → ha valore `1` ed indica che l'*Auto 3* (riga 3) è stata noleggiata all'ora `2` (colonna 2)
  - L'elemento **`I(indice)`** rappresenta l'incasso orario per il noleggio dell'auto indicata in colonna
    - ***Esempio***: `I(2)` → indica che l'*Auto 2* (colonna 2) produce un incasso orario pari ad `8` euro
- Sia l'array che la matrice contengono esclusivamente dati numerici (evidenziati in arancio nell'esempio)

NOTA: Negli esercizi possono essere utilizzate funzioni viste a lezione (negli esempi), funzioni realizzate negli esercizi precedenti e/o funzioni built-in di MATLAB

N

<<noleggio.txt>>	1	2	3	4	5	6	7	8
	ore							
Auto 1 (indice 1)	**1**	**1**	**1**	**0**	**0**	**1**	**1**	**0**
Auto 2 (indice 2)	**0**	**0**	**1**	**1**	**0**	**1**	**0**	**0**
Auto 3 (indice 3)	**1**	**1**	**0**	**0**	**0**	**0**	**1**	**1**

I

<<incassi.txt>>	Auto 1	Auto 2	Auto 3
Incasso Orario	**8**	**10**	**5**

Esercizio 1

Scrivere una funzione chiamata `ore_noleggio`, che prenda come argomenti di input: la matrice `N` (*noleggio*) ed un numero intero `indice_auto`, e restituisca come argomento di output il totale delle ore in cui l'auto, avente indice `indice_auto`, è stata noleggiata

- ***Esempio:*** `ore_noleggio(N, 1)` → restituisce `5`

Possibile Soluzione

```
function [ totale_ore ] = ore_noleggio(N, indice_auto)
    totale_ore = sum(N(indice_auto, :));
end
```

N

<<noleggio.txt>>	1	2	3	4	5	6	7	8
Auto 1 (indice 1)	**1**	**1**	**1**	**0**	**0**	**1**	**1**	**0**
Auto 2 (indice 2)	**0**	**0**	**1**	**1**	**0**	**1**	**0**	**0**
Auto 3 (indice 3)	**1**	**1**	**0**	**0**	**0**	**0**	**1**	**1**

ore

I

<<incassi.txt>>	Auto 1	Auto 2	Auto 3
Incasso Orario	**8**	**10**	**5**

Esercizio 2

Scrivere una funzione chiamata `ore_vuote`, che prenda come argomenti di input la matrice `N` (*noleggio*), e restituisca come argomento di output il numero di ore in cui tutte le auto sono libere

- ***Esempio:*** `ore_vuote(N)` → restituisce `1`
  - **OSSERVAZIONI**: Il valore `1` si riferisce all'ora con indice `5` (unica ora in cui tutte le auto sono libere)

Possibile Soluzione 1/4

```
function [ num_ore_vuote ] = ore_vuote(N)
    [num_auto, num_ore] = size(N);
    num_ore_vuote = 0;

    for indice_ora = 1:num_ore
        if ~N(1, indice_ora) && ~N(2, indice_ora) && ~N(3, indice_ora)
            num_ore_vuote = num_ore_vuote + 1;
        end
    end
end
```

N

<<noleggio.txt>>	1	2	3	4	5	6	7	8
Auto 1 (indice 1)	**1**	**1**	**1**	**0**	**0**	**1**	**1**	**0**
Auto 2 (indice 2)	**0**	**0**	**1**	**1**	**0**	**1**	**0**	**0**
Auto 3 (indice 3)	**1**	**1**	**0**	**0**	**0**	**0**	**1**	**1**

ore

I

<<incassi.txt>>	Auto 1	Auto 2	Auto 3
Incasso Orario	**8**	**10**	**5**

Esercizio 2

Scrivere una funzione chiamata `ore_vuote`, che prenda come argomenti di input la matrice `N` (*noleggio*), e restituisca come argomento di output il numero di ore in cui tutte le auto sono libere

- ***Esempio:*** `ore_vuote(N)` → restituisce `1`
  - **OSSERVAZIONI**: Il valore `1` si riferisce all'ora con indice `5` (unica ora in cui tutte le auto sono libere)

Possibile Soluzione 2/4

```
function [ num_ore_vuote ] = ore_vuote(N)
    [num_auto, num_ore] = size(N);
    num_ore_vuote = 0;

    for indice_ora = 1:num_ore
        somma_ora_noleggio = 0;

        for indice_auto = 1:num_auto
            somma_ora_noleggio = somma_ora_noleggio + N(indice_auto, indice_ora);
        end

        if somma_ora_noleggio == 0
            num_ore_vuote = num_ore_vuote + 1;
        end
    end
end
```

N

<<noleggio.txt>>	1	2	3	4	5	6	7	8
Auto 1 (indice 1)	**1**	**1**	**1**	**0**	**0**	**1**	**1**	**0**
Auto 2 (indice 2)	**0**	**0**	**1**	**1**	**0**	**1**	**0**	**0**
Auto 3 (indice 3)	**1**	**1**	**0**	**0**	**0**	**0**	**1**	**1**

ore

I

<<incassi.txt>>	Auto 1	Auto 2	Auto 3
Incasso Orario	**8**	**10**	**5**

Esercizio 2

Scrivere una funzione chiamata `ore_vuote`, che prenda come argomenti di input la matrice `N` (*noleggio*), e restituisca come argomento di output il numero di ore in cui tutte le auto sono libere

- ***Esempio:*** `ore_vuote(N)` → restituisce `1`
  - **OSSERVAZIONI**: Il valore `1` si riferisce all'ora con indice `5` (unica ora in cui tutte le auto sono libere)

Possibile Soluzione 3/4

```
function [ num_ore_vuote ] = ore_vuote(N)
    [num_auto, num_ore] = size(N);
    num_ore_vuote = 0;

    for indice_ora = 1:num_ore
        if sum(N(:, indice_ora)) == 0
            num_ore_vuote = num_ore_vuote + 1;
        end
    end
end
```

N

<<noleggio.txt>>	1	2	3	4	5	6	7	8
Auto 1 (indice 1)	**1**	**1**	**1**	**0**	**0**	**1**	**1**	**0**
Auto 2 (indice 2)	**0**	**0**	**1**	**1**	**0**	**1**	**0**	**0**
Auto 3 (indice 3)	**1**	**1**	**0**	**0**	**0**	**0**	**1**	**1**

ore

I

<<incassi.txt>>	Auto 1	Auto 2	Auto 3
Incasso Orario	**8**	**10**	**5**

Esercizio 2

Scrivere una funzione chiamata `ore_vuote`, che prenda come argomenti di input la matrice `N` (*noleggio*), e restituisca come argomento di output il numero di ore in cui tutte le auto sono libere

- ***Esempio:*** `ore_vuote(N)` → restituisce `1`
  - **OSSERVAZIONI**: Il valore `1` si riferisce all'ora con indice `5` (unica ora in cui tutte le auto sono libere)

Possibile Soluzione 4/4

```
function [ num_ore_vuote ] = ore_vuote(N)
    [num_auto, num_ore] = size(N);
    array_riga_booleano = ones(1, num_ore);

    for indice_auto = 1:num_auto
        array_riga_booleano = array_riga_booleano & ~N(indice_auto, :);
    end

    num_ore_vuote = sum(array_riga_booleano);
end
```

N

<<noleggio.txt>>	1	2	3	4	5	6	7	8
Auto 1 (indice 1)	**1**	**1**	**1**	**0**	**0**	**1**	**1**	**0**
Auto 2 (indice 2)	**0**	**0**	**1**	**1**	**0**	**1**	**0**	**0**
Auto 3 (indice 3)	**1**	**1**	**0**	**0**	**0**	**0**	**1**	**1**

ore

I

<<incassi.txt>>	Auto 1	Auto 2	Auto 3
Incasso Orario	**8**	**10**	**5**

Esercizio 3

Scrivere una funzione chiamata `incassi_totali`, che prenda come argomenti di input: la matrice `N` (*noleggio*) e l'array `I` (*incassi*), e restituisca come argomento di output l'ammontare degli incassi totali giornalieri

- ***Esempio***: `incassi_totali(N, I)` → restituisce `90`

Possibile Soluzione 1/3

```
function [ totale_incassi ] = incassi_totali(N, I)
    totale_incassi = sum(I * N);
end
```

Possibile Soluzione 2/3

```
function [ totale_incassi ] = incassi_totali(N, I)
    [num_auto, num_ore] = size(N);
    totale_incassi = 0;

    for indice_ora = 1:num_ore
        totale_incassi = totale_incassi + sum(N(:, indice_ora) .* I');
    end
end
```

N

<<noleggio.txt>>	1	2	3	4	5	6	7	8
Auto 1 (indice 1)	**1**	**1**	**1**	**0**	**0**	**1**	**1**	**0**
Auto 2 (indice 2)	**0**	**0**	**1**	**1**	**0**	**1**	**0**	**0**
Auto 3 (indice 3)	**1**	**1**	**0**	**0**	**0**	**0**	**1**	**1**

ore

I

<<incassi.txt>>	Auto 1	Auto 2	Auto 3
Incasso Orario	**8**	**10**	**5**

Esercizio 3

Scrivere una funzione chiamata `incassi_totali`, che prenda come argomenti di input: la matrice `N` (*noleggio*) e l'array `I` (*incassi*), e restituisca come argomento di output l'ammontare degli incassi totali giornalieri

- ***Esempio***: `incassi_totali(N, I)` → restituisce `90`

Possibile Soluzione 3/3

```
function [ totale_incassi ] = incassi_totali(N, I)
    [num_auto, num_ore] = size(N);
    somma = 0;

    for indice_ora = 1:num_ore
        for indice_auto = 1:num_auto
            somma = somma + (N(indice_auto, indice_ora) * I(indice_auto));
        end
    end

    totale_incassi = somma;
end
```

N

<<noleggio.txt>>	1	2	3	4	5	6	7	8
Auto 1 (indice 1)	**1**	**1**	**1**	**0**	**0**	**1**	**1**	**0**
Auto 2 (indice 2)	**0**	**0**	**1**	**1**	**0**	**1**	**0**	**0**
Auto 3 (indice 3)	**1**	**1**	**0**	**0**	**0**	**0**	**1**	**1**

ore

I

<<incassi.txt>>	Auto 1	Auto 2	Auto 3
Incasso Orario	**8**	**10**	**5**

Esercizio 4

Scrivere una funzione chiamata `incassi_non_conseguiti`, che prenda come argomenti di input: la matrice `N` (noleggio) e l'array `I` (*incassi*), e restituisca come output gli **incassi totali giornalieri non conseguiti**. Si supponga che gli incassi totali non conseguiti sono ottenuti come: ***incassi_potenziali – incassi_totali***, dove ***incassi_potenziali*** sono gli incassi giornalieri che si sarebbero ottenuti se tutte le auto fossero state noleggiate durante tutte le ore

- ***Esempio:*** `incassi_non_conseguiti(N, I)` → restituisce `94`

Possibile Soluzione 1/3

```
function [ totale_inc_non_conseguiti ] = incassi_non_conseguiti(N, I)
    totale_inc_non_conseguiti = incassi_totali(~N, I);
end
```

N

<<noleggio.txt>>	ore 1	2	3	4	5	6	7	8
Auto 1 (indice 1)	**1**	**1**	**1**	**0**	**0**	**1**	**1**	**0**
Auto 2 (indice 2)	**0**	**0**	**1**	**1**	**0**	**1**	**0**	**0**
Auto 3 (indice 3)	**1**	**1**	**0**	**0**	**0**	**0**	**1**	**1**

I

<<incassi.txt>>	Auto 1	Auto 2	Auto 3
Incasso Orario	**8**	**10**	**5**

Esercizio 4

Scrivere una funzione chiamata `incassi_non_conseguiti`, che prenda come argomenti di input: la matrice `N` (noleggio) e l'array `I` (*incassi*), e restituisca come output gli **incassi totali giornalieri non conseguiti**. Si supponga che gli incassi totali non conseguiti sono ottenuti come: ***incassi_potenziali – incassi_totali***, dove ***incassi_potenziali*** sono gli incassi giornalieri che si sarebbero ottenuti se tutte le auto fossero state noleggiate durante tutte le ore

- ***Esempio:*** `incassi_non_conseguiti(N, I)` → restituisce `94`

Possibile Soluzione 2/3

```
function [ totale_inc_non_conseguiti ] = incassi_non_conseguiti(N, I)
    [num_auto, num_ore] = size(N);
    incassi_potenziali = 0;

    for indice_auto = 1:num_auto
        incassi_potenziali = incassi_potenziali + (num_ore * I(indice_auto));
    end

    totale_inc_non_conseguiti = incassi_potenziali - incassi_totali(N, I);
end
```

N

<<noleggio.txt>>	*ore* 1	2	3	4	5	6	7	8
Auto 1 (indice 1)	**1**	**1**	**1**	**0**	**0**	**1**	**1**	**0**
Auto 2 (indice 2)	**0**	**0**	**1**	**1**	**0**	**1**	**0**	**0**
Auto 3 (indice 3)	**1**	**1**	**0**	**0**	**0**	**0**	**1**	**1**

I

<<incassi.txt>>	Auto 1	Auto 2	Auto 3
Incasso Orario	**8**	**10**	**5**

Esercizio 4

Scrivere una funzione chiamata `incassi_non_conseguiti`, che prenda come argomenti di input: la matrice `N` (noleggio) e l'array `I` (*incassi*), e restituisca come output gli **incassi totali giornalieri non conseguiti**. Si supponga che gli incassi totali non conseguiti sono ottenuti come: ***incassi_potenziali – incassi_totali***, dove ***incassi_potenziali*** sono gli incassi giornalieri che si sarebbero ottenuti se tutte le auto fossero state noleggiate durante tutte le ore

- ***Esempio:*** `incassi_non_conseguiti(N, I)` → restituisce `94`

Possibile Soluzione 3/3

```
function [ totale_inc_non_conseguiti ] = incassi_non_conseguiti(N, I)
    [num_auto, num_ore] = size(N);
    incassi_potenziali = 0;

    for indice_auto = 1:num_auto
        incassi_potenziali = incassi_potenziali + (num_ore * I(indice_auto));
    end

    incassi_totali_giornalieri = 0;

    for indice_auto = 1:num_auto
        incassi_totali_giornalieri = incassi_totali_giornalieri + (sum(N(indice_auto, :)) * I(indice_auto));
    end

    totale_inc_non_conseguiti = incassi_potenziali - incassi_totali_giornalieri;
end
```

N

	ore							
<<noleggio.txt>>	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Auto 1 (indice 1)	**1**	**1**	**1**	**0**	**0**	**1**	**1**	**0**
Auto 2 (indice 2)	**0**	**0**	**1**	**1**	**0**	**1**	**0**	**0**
Auto 3 (indice 3)	**1**	**1**	**0**	**0**	**0**	**0**	**1**	**1**

I

<<incassi.txt>>	Auto 1	Auto 2	Auto 3
Incasso Orario	**8**	**10**	**5**

Esercizio 5

Scrivere un M-File Script chiamato `noleggio_auto_script.m` che effettui le seguenti operazioni

1. Importi la matrice `N` dal file `noleggio.txt`
2. Importi la matrice `I` dal file `incassi.txt`
3. Invochi la funzione dell'Esercizio 4 (chiamata `incassi_non_conseguiti`) con gli argomenti di input: `N` e `I`, ed infine mostri a video il risultato della funzione stessa

NOTA: I file `noleggio.txt` e `incassi.txt` contengono solo dati numerici. È utilizzato il separatore virgola (**,**) per separare le colonne (***suggerimento:*** utilizzare la funzione `importdata`). Si assuma che i file siano memorizzati all'interno della **Current Directory**

Possibile Soluzione

```
N = importdata('noleggio.txt');
I = importdata('incassi.txt');

incassi_non_conseguiti(N, I)
```

Contenuto del file
`noleggio_auto_script.m`

N *ore*

<<noleggio.txt>>	1	2	3	4	5	6	7	8
Auto 1 (indice 1)	**1**	**1**	**1**	**0**	**0**	**1**	**1**	**0**
Auto 2 (indice 2)	**0**	**0**	**1**	**1**	**0**	**1**	**0**	**0**
Auto 3 (indice 3)	**1**	**1**	**0**	**0**	**0**	**0**	**1**	**1**

I

<<incassi.txt>>	Auto 1	Auto 2	Auto 3
Incasso Orario	**8**	**10**	**5**

Esercizio 6

Scrivere una funzione chiamata `grafico_noleggio`, che prenda come argomenti di input: la matrice `N` (*noleggio*), e mostri un grafico con le seguenti proprietà:

- Sull'asse *X*, riporti gli indici di tutte le ore
- Sull'asse *Y*, per ogni punto, riporti il numero di auto noleggiate nell'ora riportata sull'asse *X*
- *Titolo:* 'Grafico Noleggio'
- *Etichetta Asse X:* 'Ore'
- *Etichetta Asse Y*: 'Auto Noleggiate'

Inoltre, restituisca un array contenente i valori assegnati all'asse *Y* (***Esempio***: Array riga o colonna `[ 2 2 2 1 0 2 2 1 ]`)

Possibile Soluzione

```
function [ auto_noleggiate ] = grafico_noleggio(N)
    [num_auto, num_ore] = size(N);

    x = 1:num_ore;
    y = sum(N);

    plot(x, y);
    title('Grafico Noleggio');
    xlabel('Ore');
    ylabel('Auto Noleggiate');

    auto_noleggiate = y;
end
```