Università di Salerno
Dipartimento di Ingegneria Industriale

UNIVERSITÀ DEGLI STUDI DI SALERNO

Fondamenti di Informatica

Programmazione in MATLAB | Gestione File | Grafici
Esercitazione Per Casa – 1 | Possibili Soluzioni

Prof. Raffaele Pizzolante

A.A. 2016/17

MATLAB

P *sedie*

posti	1	2	3	4	5
Fila 1 (indice 1)	1	0	1	1	0
Fila 2 (indice 2)	0	1	0	0	1
Fila 3 (indice 3)	0	1	0	1	0
Fila 4 (indice 4)	1	1	1	0	0
Fila 5 (indice 5)	1	1	0	0	1
Fila 6 (indice 6)	1	1	0	1	0

T *sedie*

tipo_posti	1	2	3	4	5
Fila 1 (indice 1)	1	1	1	1	1
Fila 2 (indice 2)	1	1	1	1	1
Fila 3 (indice 3)	1	2	2	2	1
Fila 4 (indice 4)	2	2	2	2	2
Fila 5 (indice 5)	2	2	3	2	2
Fila 6 (indice 6)	3	3	3	3	3

I

Incassi_teatro	Tipo 1 (ind. 1)	Tipo 2 (ind. 2)	Tipo 3 (ind. 3)
Incasso Singolo	**6**	**7.5**	**8**

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

- In questa esercitazione verrà utilizzata una matrice **`P`**, una matrice **`T`** ed un array riga **`I`**
  - La matrice **`P`** rappresenta **i posti liberi ed occupati** all'interno di una sala teatrale
    - **`P(indice_fila, indice_sedia)`** assume valore `1` se il posto `P(indice_fila, indice_sedia)` è occupato, `0` altrimenti.
      - ***Esempio***: `P(4,2)` → ha valore `1` ed indica che il posto (*Fila* `4`, *Sedia* `2`) è occupato
  - La matrice **`T`** specifica, per ogni posto, la relativa tipologia
    - ***Esempio***: `T(6,3)` → ha valore `3` ed indica che il posto (*Fila* `6`, *Sedia* `3`) è di tipo `3`
  - L'elemento **`I(indice)`** rappresenta l'<u>incasso unitario</u> (relativo a ciascun posto) per la visione di uno spettacolo, in base alla tipologia di posto occupato
    - ***Esempio***: `I(2)` → indica che l'incasso unitario di un posto, appartenente alla tipologia 2 (colonna 2) è pari a `7.5` euro
  - Link Script Matrici Esempio:
    - http://www.di.unisa.it/dottorandi/pizzolante/FI_201617/materiale/matlab/aggiuntivo/Lezione_22_ScriptMatrici.txt
- Sia l'array che le matrice contengono esclusivamente dati numerici (evidenziati in arancio nell'esempio)

P

posti	1	2	3	4	5
Fila 1 (indice 1)	1	0	1	1	0
Fila 2 (indice 2)	0	1	0	0	1
Fila 3 (indice 3)	0	1	0	1	0
Fila 4 (indice 4)	1	1	1	0	0
Fila 5 (indice 5)	1	1	0	0	1
Fila 6 (indice 6)	1	1	0	1	0

sedie

T

tipo_posti	1	2	3	4	5
Fila 1 (indice 1)	1	1	1	1	1
Fila 2 (indice 2)	1	1	1	1	1
Fila 3 (indice 3)	1	2	2	2	1
Fila 4 (indice 4)	2	2	2	2	2
Fila 5 (indice 5)	2	2	3	2	2
Fila 6 (indice 6)	3	3	3	3	3

sedie

I

Incassi_teatro	Tipo 1 (ind. 1)	Tipo 2 (ind. 2)	Tipo 3 (ind. 3)
Incasso Singolo	6	7.5	8

Esercizio 1

Scrivere una funzione chiamata `incassi_totali_teatro`, che prenda come argomenti di input: la matrice `P` (*posti*), la matrice `T` (*tipo_posti*) e l'array `I` (*incassi*), e restituisca, come argomento di output, il totale degli incassi ottenuti

- ***Esempio:*** `incassi_totali_teatro(P, T, I)` → restituisce `114`

Possibile Soluzione

```
function [ incassi_totali ] = incassi_totali_teatro(P, T, I)
    [num_file, num_sedie] = size(P);
    somma = 0;

    for indice_fila = 1:num_file
        for indice_sedia = 1:num_sedie
            somma = somma + (P(indice_fila, indice_sedia) * I(T(indice_fila, indice_sedia)));
        end
    end

    incassi_totali = somma;
end
```

P *sedie*

posti	**1**	**2**	**3**	**4**	**5**
Fila 1 (indice 1)	**1**	**0**	**1**	**1**	**0**
Fila 2 (indice 2)	**0**	**1**	**0**	**0**	**1**
Fila 3 (indice 3)	**0**	**1**	**0**	**1**	**0**
Fila 4 (indice 4)	**1**	**1**	**1**	**0**	**0**
Fila 5 (indice 5)	**1**	**1**	**0**	**0**	**1**
Fila 6 (indice 6)	**1**	**1**	**0**	**1**	**0**

T *sedie*

tipo_posti	**1**	**2**	**3**	**4**	**5**
Fila 1 (indice 1)	**1**	**1**	**1**	**1**	**1**
Fila 2 (indice 2)	**1**	**1**	**1**	**1**	**1**
Fila 3 (indice 3)	**1**	**2**	**2**	**2**	**1**
Fila 4 (indice 4)	**2**	**2**	**2**	**2**	**2**
Fila 5 (indice 5)	**2**	**2**	**3**	**2**	**2**
Fila 6 (indice 6)	**3**	**3**	**3**	**3**	**3**

I

Incassi_teatro	Tipo 1 (ind. 1)	Tipo 2 (ind. 2)	Tipo 3 (ind. 3)
Incasso Singolo	**6**	**7.5**	**8**

Esercizio 2

Scrivere una funzione chiamata `numero_posti_tipo`, che prenda come argomenti di input la matrice `P` (*posti*), la matrice `T` (*tipo_posti*) ed un intero `indice_tipo`, e restituisca come argomento di output il numero di **posti occupati** del tipo, avente come indice `indice_tipo`

- ***Esempio:*** `numero_posti_tipo(P, T, 1)` → restituisce `5`

Possibile Soluzione 1/3

```
function [ numero_posti ] = numero_posti_tipo(P, T, indice_tipo)
    numero_posti = length(find(P .* T == indice_tipo));
end
```

P *sedie*

posti	**1**	**2**	**3**	**4**	**5**
Fila 1 (indice 1)	**1**	**0**	**1**	**1**	**0**
Fila 2 (indice 2)	**0**	**1**	**0**	**0**	**1**
Fila 3 (indice 3)	**0**	**1**	**0**	**1**	**0**
Fila 4 (indice 4)	**1**	**1**	**1**	**0**	**0**
Fila 5 (indice 5)	**1**	**1**	**0**	**0**	**1**
Fila 6 (indice 6)	**1**	**1**	**0**	**1**	**0**

T *sedie*

tipo_posti	**1**	**2**	**3**	**4**	**5**
Fila 1 (indice 1)	**1**	**1**	**1**	**1**	**1**
Fila 2 (indice 2)	**1**	**1**	**1**	**1**	**1**
Fila 3 (indice 3)	**1**	**2**	**2**	**2**	**1**
Fila 4 (indice 4)	**2**	**2**	**2**	**2**	**2**
Fila 5 (indice 5)	**2**	**2**	**3**	**2**	**2**
Fila 6 (indice 6)	**3**	**3**	**3**	**3**	**3**

I

Incassi_teatro	Tipo 1 (ind. 1)	Tipo 2 (ind. 2)	Tipo 3 (ind. 3)
Incasso Singolo	**6**	**7.5**	**8**

Esercizio 2

Scrivere una funzione chiamata `numero_posti_tipo`, che prenda come argomenti di input la matrice `P` (*posti*), la matrice `T` (*tipo_posti*) ed un intero `indice_tipo`, e restituisca come argomento di output il <u>numero di **posti occupati**</u> del tipo, avente come indice `indice_tipo`

- ***Esempio:*** `numero_posti_tipo(P, T, 1)` → restituisce `5`

Possibile Soluzione 2/3

```
function [ numero_posti ] = numero_posti_tipo(P, T, indice_tipo)
    [num_file, num_sedie] = size(P);
    s = 0;

    for indice_fila = 1:num_file
        for indice_sedia = 1:num_sedie
            if P(indice_fila, indice_sedia) == 1 && T(indice_fila, indice_sedia) == indice_tipo
                s = s + 1;
            end
        end
    end

    numero_posti = s;
end
```

P *sedie*

posti	**1**	**2**	**3**	**4**	**5**
Fila 1 (indice 1)	**1**	**0**	**1**	**1**	**0**
Fila 2 (indice 2)	**0**	**1**	**0**	**0**	**1**
Fila 3 (indice 3)	**0**	**1**	**0**	**1**	**0**
Fila 4 (indice 4)	**1**	**1**	**1**	**0**	**0**
Fila 5 (indice 5)	**1**	**1**	**0**	**0**	**1**
Fila 6 (indice 6)	**1**	**1**	**0**	**1**	**0**

T *sedie*

tipo_posti	**1**	**2**	**3**	**4**	**5**
Fila 1 (indice 1)	**1**	**1**	**1**	**1**	**1**
Fila 2 (indice 2)	**1**	**1**	**1**	**1**	**1**
Fila 3 (indice 3)	**1**	**2**	**2**	**2**	**1**
Fila 4 (indice 4)	**2**	**2**	**2**	**2**	**2**
Fila 5 (indice 5)	**2**	**2**	**3**	**2**	**2**
Fila 6 (indice 6)	**3**	**3**	**3**	**3**	**3**

I

Incassi_teatro	Tipo 1 (ind. 1)	Tipo 2 (ind. 2)	Tipo 3 (ind. 3)
Incasso Singolo	**6**	**7.5**	**8**

Esercizio 2

Scrivere una funzione chiamata `numero_posti_tipo`, che prenda come argomenti di input la matrice `P` (*posti*), la matrice `T` (*tipo_posti*) ed un intero `indice_tipo`, e restituisca come argomento di output il <u>numero di **posti occupati**</u> del tipo, avente come indice `indice_tipo`

- ***Esempio:*** `numero_posti_tipo(P, T, 1)` → restituisce `5`

Possibile Soluzione 3/3

```
function [ numero_posti ] = numero_posti_tipo(P, T, indice_tipo)
    [num_file, num_sedie] = size(P);
    s = 0;

    for indice_fila = 1:num_file
        for indice_sedia = 1:num_sedie
            if P(indice_fila, indice_sedia) == 1
                if T(indice_fila, indice_sedia) == indice_tipo
                    s = s + 1;
                end
            end
        end
    end

    numero_posti = s;
end
```

P *sedie*

posti	**1**	**2**	**3**	**4**	**5**
Fila 1 (indice 1)	**1**	**0**	**1**	**1**	**0**
Fila 2 (indice 2)	**0**	**1**	**0**	**0**	**1**
Fila 3 (indice 3)	**0**	**1**	**0**	**1**	**0**
Fila 4 (indice 4)	**1**	**1**	**1**	**0**	**0**
Fila 5 (indice 5)	**1**	**1**	**0**	**0**	**1**
Fila 6 (indice 6)	**1**	**1**	**0**	**1**	**0**

T *sedie*

tipo_posti	**1**	**2**	**3**	**4**	**5**
Fila 1 (indice 1)	**1**	**1**	**1**	**1**	**1**
Fila 2 (indice 2)	**1**	**1**	**1**	**1**	**1**
Fila 3 (indice 3)	**1**	**2**	**2**	**2**	**1**
Fila 4 (indice 4)	**2**	**2**	**2**	**2**	**2**
Fila 5 (indice 5)	**2**	**2**	**3**	**2**	**2**
Fila 6 (indice 6)	**3**	**3**	**3**	**3**	**3**

I

Incassi_teatro	Tipo 1 (ind. 1)	Tipo 2 (ind. 2)	Tipo 3 (ind. 3)
Incasso Singolo	**6**	**7.5**	**8**

Esercizio 3

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

1. Importi la matrice `P` dal file `posti.txt`
2. Importi la matrice `T` dal file `tipo_posti.txt`
3. Importi la matrice `I` dal file `incassi_teatro.txt`
4. Invochi la funzione dell'Esercizio 1 (chiamata `incassi_totali_teatro`) con gli argomenti di input: `P`, `T` ed `I`, e mostri a video il risultato della funzione stessa
5. Generi un grafico a barre con le seguenti caratteristiche
   1. Asse *X* → Rappresenta gli indici dei tipi di posto
   2. Asse *Y* → Ad ogni punto di *Y*, rappresenti l'ammontare dell'incasso singolo, in base al tipo (specificato sull'asse *X*)

NOTA: I file `posti.txt`, `tipo_posti.txt` ed `incassi_teatro.txt` (mostrati in seguito) 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**

P

posti	1	2	3	4	5
Fila 1 (indice 1)	1	0	1	1	0
Fila 2 (indice 2)	0	1	0	0	1
Fila 3 (indice 3)	0	1	0	1	0
Fila 4 (indice 4)	1	1	1	0	0
Fila 5 (indice 5)	1	1	0	0	1
Fila 6 (indice 6)	1	1	0	1	0

sedie

T

tipo_posti	1	2	3	4	5
Fila 1 (indice 1)	1	1	1	1	1
Fila 2 (indice 2)	1	1	1	1	1
Fila 3 (indice 3)	1	2	2	2	1
Fila 4 (indice 4)	2	2	2	2	2
Fila 5 (indice 5)	2	2	3	2	2
Fila 6 (indice 6)	3	3	3	3	3

sedie

I

Incassi_teatro	Tipo 1 (ind. 1)	Tipo 2 (ind. 2)	Tipo 3 (ind. 3)
Incasso Singolo	6	7.5	8

Esercizio 3

Possibile Soluzione (contenuto del file `teatro_script.m`)

```
P = importdata('posti.txt');
T = importdata('tipo_posti.txt');
I = importdata('incassi_teatro.txt');

incassi_totali_teatro(P, T, I)

x = 1:1:length(I);
y = I;

bar(x, y)
```