



UNIVERSITÀ DEGLI STUDI DI SALERNO

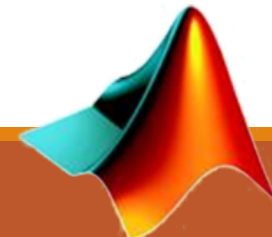


Fondamenti di Informatica

Strutture Selettive, Iterative, Gestione File e
Grafici in MATLAB: Esercizi (Magazzini)

Prof. Arcangelo Castiglione

A.A. 2016/17



MATLAB

	<<magazzini.txt>>	Pantalone (ind. 1)	Camicia (ind. 2)	Maglia (ind. 3)
M	Roma (indice 1)	4	25	18
	Milano (indice 2)	15	3	19
	Napoli (indice 3)	20	10	21
	<<prezzi.txt>>	Pantalone	Camicia	Maglia
P	Prezzo	40	30	20

- In questa prova verranno utilizzati una matrice **M** ed un array **P**
 - Ogni cella di **M** rappresenta la quantità di un certo prodotto (colonna) disponibile in una determinata sede del magazzino (riga)
 - Ogni cella di **P** rappresenta l'importo necessario per uno specifico prodotto (colonna)
- La matrice **M** contiene esclusivamente dati numerici (evidenziati in arancio nell'esempio)
- L'array **P** contiene esclusivamente dati numerici (evidenziati in arancio nell'esempio)

	<<magazzini.txt>>	Pantalone (ind. 1)	Camicia (ind. 2)	Maglia (ind. 3)
M	Roma (indice 1)	4	25	18
	Milano (indice 2)	15	3	19
	Napoli (indice 3)	20	10	21
	<<prezzi.txt>>	Pantalone	Camicia	Maglia
P	Prezzo	40	30	20

Esercizio 1

- Scrivere una funzione `scorte`, che prenda in input la matrice **M** (*magazzini*) e restituisca un array **S**, in cui ogni elemento **S** (**i**) è definito come segue

$$S(i) = \begin{cases} 1 & \text{se è necessario fare un rifornimento nella sede con indice } i \\ 0 & \text{altrimenti} \end{cases}$$

- Si noti che **un rifornimento è necessario** se vi sono uno o più prodotti **la cui quantità** presente in magazzino è **inferiore a 5**

- Nell'esempio, la funzione `scorte` restituirà $S = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$

- **Nota:** È indifferente che **S** sia un vettore riga o un vettore colonna

		<<magazzini.txt>>	Pantalone (ind. 1)	Camicia (ind. 2)	Maglia (ind. 3)
M	Roma (indice 1)		4	25	18
	Milano (indice 2)		15	3	19
	Napoli (indice 3)		20	10	21
		<<prezzi.txt>>	Pantalone	Camicia	Maglia
P	Prezzo		40	30	20

Esercizio 1 - Possibile Soluzione

```
function [S] = scorte(M)
    [nr, nc] = size(M);

    for i=1:nr
        min_sede = min(M(:,i));

        if (min_sede < 5)
            S(i) = 1;
        else
            S(i) = 0;
        end
    end
    S = S';
end
```

Esempio d'uso

```
>> S = scorte(M)

S =

     1
     1
     0
```

	<<magazzini.txt>>	Pantalone (ind. 1)	Camicia (ind. 2)	Maglia (ind. 3)
M	Roma (indice 1)	4	25	18
	Milano (indice 2)	15	3	19
	Napoli (indice 3)	20	10	21
	<<prezzi.txt>>	Pantalone	Camicia	Maglia
P	Prezzo	40	30	20

Esercizio 2

- Scrivere una funzione chiamata `importi`, che prenda in input la matrice **M** (*magazzini*) e l'array **P** (*prezzi*), e restituisca in output un array **I**
- Ogni elemento dell'array **I** dovrà contenere l'importo totale ricavabile dalla vendita della merce contenuta in un determinato magazzino
- **Esempio**
 - $I(1) = (4 * 40) + (25 * 30) + (18 * 20) \rightarrow$ magazzino **Roma**
 - $I(2) = (15 * 40) + (3 * 30) + (19 * 20) \rightarrow$ magazzino **Milano**
 - $I(3) = (20 * 40) + (10 * 30) + (21 * 20) \rightarrow$ magazzino **Napoli**
- **Nota:** È indifferente che **I** sia un vettore riga o un vettore colonna

<<magazzini.txt>>		Pantalone (ind. 1)	Camicia (ind. 2)	Maglia (ind. 3)
M	Roma (indice 1)	4	25	18
	Milano (indice 2)	15	3	19
	Napoli (indice 3)	20	10	21
<<prezzi.txt>>		Pantalone	Camicia	Maglia
P	Prezzo	40	30	20

Esercizio 2 - Possibile Soluzione

```
function [I] = importi(M, P)
    [nr, nc] = size(M);

    for i = 1:nr
        I(i) = sum(M(i,:) .* P);
    end

    I = I';
end
```

Esempio d'uso

```
>> I = importi(M, P)

I =

    1270
    1070
    1520
```

	<<magazzini.txt>>	Pantalone (ind. 1)	Camicia (ind. 2)	Maglia (ind. 3)
M	Roma (indice 1)	4	25	18
	Milano (indice 2)	15	3	19
	Napoli (indice 3)	20	10	21
	<<prezzi.txt>>	Pantalone	Camicia	Maglia
P	Prezzo	40	30	20

Esercizio 3

- Scrivere una funzione chiamata `piu_fornito`, che prenda in input la matrice **M** (*magazzini*) e restituisca in output l'indice del magazzino che ha più prodotti

Possibile Soluzione

```
function [indice_max] = piu_fornito(M)
[nr, nc] = size(M);

max = sum(M(1,:));
indice_max = 1;

for i=2:nr
    if max < sum(M(i,:))
        max = sum(M(i,:));
        indice_max = i;
    end
end
end
```

Esempio d'uso

```
>> indice_piu_fornito = piu_fornito(M)

indice_piu_fornito =

    3
```

<<magazzini.txt>>		Pantalone (ind. 1)	Camicia (ind. 2)	Maglia (ind. 3)
M	Roma (indice 1)	4	25	18
	Milano (indice 2)	15	3	19
	Napoli (indice 3)	20	10	21
<<prezzi.txt>>		Pantalone	Camicia	Maglia
P	Prezzo	40	30	20

Esercizio 4

- Scrivere una funzione chiamata `totale_camicie`, che prenda in input la matrice **M** (*magazzini*) e restituisca in output il **numero totale di camicie** (in tutti i magazzini)

Possibile Soluzione

```
function [num_camicie] = totale_camicie(M)
    num_camicie = sum(M(:,2));
end
```

Esempio d'uso

```
>> camicie = totale_camicie(M)

camicie =
    38
```


	<<magazzini.txt>>	Pantalone (ind. 1)	Camicia (ind. 2)	Maglia (ind. 3)
M	Roma (indice 1)	4	25	18
	Milano (indice 2)	15	3	19
	Napoli (indice 3)	20	10	21
	<<prezzi.txt>>	Pantalone	Camicia	Maglia
P	Prezzo	40	30	20

Esercizio 5

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

1. Importi la matrice `M` dal file `magazzini.txt`
2. Importi l'array `P` dal file `prezzi.txt`
3. Invochi la funzione dell'Esercizio 3 (chiamata `piu_fornito`) con argomento di input `M`, e mostri a video l'output della funzione stessa

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

Possibile Soluzione

```
M = importdata('magazzini.txt');
P = importdata('prezzi.txt');

piu_fornito(M)
```