



UNIVERSITÀ DEGLI STUDI DI SALERNO

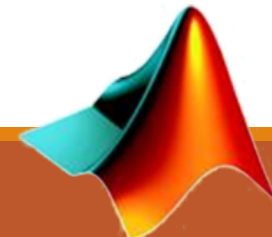


Fondamenti di Informatica

Strutture Selettive, Iterative e Gestione File in
MATLAB: Esercitazione 2 – Possibili Soluzioni

Prof. Arcangelo Castiglione

A.A. 2016/17



MATLAB®

Esercizio 1

- Scrivere una funzione che prenda in input una matrice A e si comporti in maniera identica al comando `sum(A)` fornito da MATLAB
- Scrivere una funzione che prenda in input una matrice A e si comporti in maniera identica al comando `sum(A, 2)` fornito da MATLAB

N.B. Si consiglia l'uso del ciclo `for`

Esercizio 1

(Possibili Soluzioni)

```
% Funzione per simulare sum(A)
function [ sum_A ] = miaSomma(A)
    [m,n] = size(A);
    for c = 1:n
        x = 0;
        for r = 1:m
            x = x + A(r,c);
        end
        sum_A(c) = x;
    end
end
```

Esercizio 1

(Possibili Soluzioni)

```
% Funzione per simulare sum(A, 2)
function [ sum_A ] = miaSomma2(A)
    [m,n] = size(A);
    for c = 1:m
        x = 0;
        for r = 1:n
            x = x + A(c,r);
        end
        sum_A(c) = x;
    end
    sum_A = sum_A';
end
```

Esercizio 2

- Scrivere una funzione chiamata `mia_trasposta` che prenda in input una matrice **A** e restituisca in output la relativa matrice trasposta

$$A = \begin{bmatrix} 1 & 9 & 6 \\ 3 & 2 & 21 \\ -2 & 6 & 8 \end{bmatrix} \quad \longrightarrow \quad \text{trasposta}(A) = \begin{bmatrix} 1 & 3 & -2 \\ 9 & 2 & 6 \\ 6 & 21 & 8 \end{bmatrix}$$

N.B. Si consiglia l'uso del ciclo [for](#)

Esercizio 2

(Possibile Soluzione)

```
function [t] = mia_trasposta(m)
    [r,c]=size(m);
    for i=1:r
        for j=1:c
            t(j,i)=m(i,j);
        end
    end
end
```

Esercizio 3 (MCD – Divisioni Successive)

Massimo Comun Divisore (MCD)

- Siano **A** e **B** due interi, non nulli (> 0), allora il **MCD** fra **A** e **B** è definito come il **più grande intero positivo che divide A e B**
- Si scriva una funzione che prenda in input **A** e **B** e restituisca il relativo MCD

Algoritmo Euclideo per il calcolo dell'MCD (basato su divisioni successive)

- Se **A** < **B**
 - Scambia **A** e **B**
- **resto** = $\text{mod}(\mathbf{A}, \mathbf{B})$ % $\text{mod}(\mathbf{A}, \mathbf{B}) \rightarrow$ Resto della divisione euclidea tra **A** e **B**
- Finché **resto** è diverso da 0
 - **A** = **B**
 - **B** = **resto**
 - **resto** = $\text{mod}(\mathbf{A}, \mathbf{B})$
- Restituisci **B**

Esempi d'uso

- $\text{MCD}(9, 3) = 3$
- $\text{MCD}(11, 2) = 1$

Confrontare l'output della funzione con quello restituito dalla funzione **gcd** di MATLAB



Esercizio 3 (MCD – Divisioni Successive)

(Possibile Soluzione)

```
function gcd = mcd(n,m)
    if n < m
        tmp = n;
        n = m;
        m = tmp;
    end
    r = mod(n,m);
    while r ~= 0
        n = m;
        m = r;
        r = mod(n,m);
    end
    gcd=m;
end
```


Esercizio 4 (MCD – Sottrazioni Successive)

Massimo Comun Divisore (MCD)

- Siano **A** e **B** due interi, non nulli (> 0), allora il **MCD** fra **A** e **B** è definito come il **più grande intero positivo che divide A e B**
- Si scriva una funzione che prenda in input **A** e **B** e restituisca il relativo MCD

Algoritmo Euclideo per il calcolo dell'MCD (basato su divisioni successive)

- Se **A** < **B**
 - Scambia **A** e **B**
- Finché **A** è diverso da **B**
 - Se **A** > **B**
 - **A** = **A** - **B**
 - Else
 - **B** = **B** - **A**
- Restituisci **A**

Esempi d'uso

- $\text{MCD}(9, 3) = 3$

- $\text{MCD}(11, 2) = 1$

Confrontare l'output della funzione con quello restituito dalla funzione `gcd` di MATLAB



Esercizio 4 (MCD – Divisioni Successive)

(Possibile Soluzione)

```
function gcd = mcd2(n,m)
    if n < m
        tmp = n;
        n = m;
        m = tmp;
    end
    while n ~= m
        if n > m
            n = n-m;
        else
            m = m-n;
        end
    end
    gcd=n;
end
```

Esercizio 5 (minimo comune multiplo)

(Metodo 1)

- Il minimo comune multiplo (*mcm*) tra due numeri a e b può essere ottenuto mediante la fattorizzazione in primi di ciascun numero

$$a = p_1^{a_1} \cdot \dots \cdot p_n^{a_n}$$

$$b = p_1^{b_1} \cdot \dots \cdot p_n^{b_n}$$

- Dove i p_i sono tutti i fattori primi di a e b . Se p_i non compare in una fattorizzazione, allora il relativo esponente è considerato 0. Il *mcm* può essere calcolato attraverso la formula seguente

$$mcm(a, b) = \prod_{i=1}^n p_i^{\max(a_i, b_i)}$$

- Per esempio, si consideri $mcm(12, 30)$

$$12 = 2^2 \cdot 3^1 \cdot 5^0$$

$$30 = 2^1 \cdot 3^1 \cdot 5^1$$

$$mcm(12, 30) = 2^2 \cdot 3^1 \cdot 5^1 = 60$$

Esercizio 6 (minimo comune multiplo) (Metodo 2)

- Sfruttando i risultati derivanti dal **Teorema Fondamentale dell'aritmetica** (*Corollario del Primo Teorema di Euclide*), il minimo comune multiplo (*mcm*) tra due numeri può essere calcolato, utilizzando il massimo comune divisore (*mcd*), mediante la seguente formula

$$mcm(a, b) = \frac{b \cdot a}{mcd(b, a)}$$

È consigliato l'utilizzo delle funzioni create per l'esercizio precedente

Per entrambi i metodi dell'Esercizio 5, confrontare l'output della funzione con quello restituito dalla funzione `lcm` di MATLAB

Esercizio 6 (minimo comune multiplo)

(Metodo 2 – Possibile Soluzione)

```
function lcm = mcm(n,m)

    lcm = (m * n) / gcd(m,n);

end
```

Esempio A →

M = 4

N = 3

	Camera1	Camera2	Camera3	Camera4
Piano 1	3	2	0	1
Piano 2	1	3	2	2
Piano 3	1	3	0	0

Esercizio 7

- **Ospiti Albergo (Parte 1)**

- Nei seguenti esercizi, le funzioni richiederanno in input una generica matrice \mathbf{A} , di dimensione $\mathbf{M} \times \mathbf{N}$, che rappresenterà il numero di ospiti in un albergo in una certa data, per ciascuna camera (colonne) su ogni piano della struttura (righe)
 - In ogni cella di \mathbf{A} sarà contenuto il numero di ospiti di una camera specifica (0 → camera vuota)
 - Per semplicità si assuma che tutti i piani abbiano lo stesso numero di camere
- **Esercizio 7.1**
 - Scrivere una funzione che prende in input \mathbf{A} e restituisce in output il numero totale di ospiti (nell'esempio, la funzione restituirà 18)
- **Esercizio 7.2**
 - Scrivere una funzione che prende in input \mathbf{A} e restituisce in output il numero di camere libere (nell'esempio, la funzione restituirà 3)
- **Esercizio 7.3**
 - Scrivere una funzione che prende in input \mathbf{A} e individua il piano con più ospiti, restituendo inoltre il numero di ospiti stessi all'interno di tale piano (nell'esempio, la funzione restituirà 8)
- **Esercizio 7.4**
 - Scrivere una funzione che prende in input \mathbf{A} e restituisce il numero massimo di ospiti che alloggiano in una camera della struttura (nell'esempio, la funzione restituirà 3)

N.B.: Le funzioni di tali esercizi possono invocare ulteriori funzioni, sia viste a lezione che contenute negli esercizi precedenti, oppure altre funzioni da voi definite

Esercizio 7

Esempio A →

M = 4

N = 3

	Camera1	Camera2	Camera3	Camera4
Piano 1	3	2	0	1
Piano 2	1	3	2	2
Piano 3	1	3	0	0

Ospiti Albergo (Parte 2)

- **Esercizio 7.5**

- Supponendo di avere il seguente file `ospiti.txt`, memorizzato all'interno della **Current Directory**, quali sono le istruzioni necessarie per caricare il file suddetto nella matrice **A**?

```
3  2  0  1
1  3  2  2
1  3  0  0
```

ospiti.txt

N.B.: Le funzioni di tali esercizi possono invocare ulteriori funzioni, sia viste a lezione che contenute negli esercizi precedenti, oppure altre funzioni da voi definite

Esercizio 7

(Possibili Soluzioni)

- Esercizio 7.1

```
function [totale_ospiti] = numero_totale_ospiti(a)
    totale_ospiti = sum(sum(a));
end
```

- Esercizio 7.2

```
function [num_camere_libere] = numero_camere_libere(a)
    [nr, nc] = size(a);
    num_camere_libere = 0;

    for r = 1:nr
        for c = 1:nc
            if a(r, c) == 0
                num_camere_libere = num_camere_libere+1;
            end
        end
    end
end
```


Esercizio 7

(Possibili Soluzioni)

- Esercizio 7.3

```
function [piano_max_osp] = piano_max_ospiti(a)
    piano_max_osp = max(sum(a, 2));
end
```

- Esercizio 7.4

```
function [max_ospiti] = max_ospiti(a)
    max_ospiti = max(max(a));
end
```

Esercizio 7

(Possibili Soluzioni)

- Esercizio 7.5

```
>> A = importdata('ospiti.txt',' ');
```