Università di Salerno
Dipartimento di Ingegneria Industriale

UNIVERSITÀ DEGLI STUDI DI SALERNO

ESERCITAZIONE 1

Fondamenti di Informatica

Introduzione alla programmazione in MATLAB: Esercitazione 1

Prof. Christian Esposito

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

A.A. 2017/18

MATLAB

Esercizio 1 – 1/3

La Tabella 2.4 riporta i costi orari per quattro tipi di processi di fabbricazione. La tabella elenca anche il numero di ore richieste in ogni processo per produrre tre prodotti differenti. Utilizzare le matrici e Matlab per risolvere i seguenti problemi: (a) determinare il costo di ogni processo per produrre una unità del prodotto 1; (b) determinare il costo per produrre una unità di ogni prodotto; (c) se vengono prodotte 10 unità del prodotto 1, 5 unità del prodotto 2 e 7 unità del prodotto 3, calcolare il costo totale.

Tabella 2.4 Costi e tempi dei processi di fabbricazione.

		Ore richieste per produrre una unità		
Processo	**Costo orario ($)**	**Prodotto 1**	**Prodotto 2**	**Prodotto 3**
Tornitura	10	6	5	4
Rettifica	12	2	3	1
Fresatura	14	3	2	5
Saldatura	9	4	0	3

Esercizio 1 – 2/3

- Tenendo conto dello svolgimento dell'Esempio 3 della seconda esercitazione su Array e Matrici, definire le seguenti funzioni
- **<u>Punto A</u>**
  - Scrivere una funzione che
    - **<u>Prenda in input</u>**
      - La matrice `m`, che rappresenta i costi e i tempi di fabbricazione (Tabella 2.4)
      - Uno scalare `i`
    - **<u>Restituisca in output</u>**
      - Il costo di ogni processo per produrre una unità del prodotto `i`

<u>NOTA</u>: Le funzioni di tali esercizi possono invocare ulteriori funzioni sia viste a lezione e sia contenute negli esercizi precedenti oppure funzioni built-in o altre funzioni da voi definite

Esercizio 1 – 3/3

- Tenendo conto dello svolgimento dell'Esempio 3 della seconda esercitazione su Array e Matrici, definire le seguenti funzioni
- **Punto B**
  - Scrivere una funzione che
    - **Prenda in input**
      - La matrice `m`, che rappresenta i costi e i tempi di fabbricazione (Tabella 2.4)
    - **Restituisca in output**
      - Un array contenente il costo (totale) per produrre una unità di ogni prodotto

NOTA: Le funzioni di tali esercizi possono invocare ulteriori funzioni sia viste a lezione e sia contenute negli esercizi precedenti oppure funzioni built-in o altre funzioni da voi definite

Esercizio 2 – 1/2

1) Supponendo che `x = 6`, calcolare a mano i risultati delle seguenti operazioni e controllarli con Matlab.

   a) `z = (x < 10)` b) `z = (x == 10)`

   c) `z = (x >= 4)` d) `z = (x ~= 7)`

2) Calcolare a mano i risultati delle seguenti operazioni e controllarli con Matlab.

   a) `z = 6 > 3 + 8` b) `z = 6 + 3 > 8`

   c) `z = 4 > (2 + 9)` d) `z = (4 < 7) + 3`

   e) `z = 4 < 7 + 3` f) `z = (4 < 7) * 5`

   g) `z = 4 < (7 * 5)` h) `z = 2/5 >= 5`

3) Supponendo che `x = [10, -2, 6, 5, -3]` e `y = [9, -3, 2, 5, -1]`, calcolare a mano i risultati delle seguenti operazioni e controllarli con Matlab.

   a) `z = (x < 6)` b) `z = (x <= y)`

   c) `z = (x == y)` d) `z = (x ~= y)`

Esercizio 2 – 2/2

4) Dati i seguenti array x e y, utilizzare Matlab per trovare tutti gli elementi di x che sono maggiori dei corrispondenti elementi di y.

```
x = [-3, 0, 0, 2, 6, 8]     y = [-5, -2, 0, 3, 4, 10]
```

Il seguente array prezzo contiene i prezzi in lire di un determinato titolo azionario nel periodo di 10 giorni. Utilizzare Matlab per trovare il numero dei giorni in cui il prezzo è stato maggiore di 200 lire.

```
prezzo = [190, 180, 220, 210, 250, 190, 170, 210, 270, 290]
```

5) I seguenti array prezzo_A e prezzo_B contengono i prezzi in lire di due titoli azionari nel periodo di 10 giorni. Utilizzare Matlab per trovare il numero dei giorni in cui il prezzo dell'azione A è stato maggiore di quello dell'azione B.

```
prezzo_A = [190, 180, 220, 210, 250, 190, 170, 210, 270, 290]
prezzo_B = [220, 170, 200, 190, 240, 180, 160, 250, 280, 270]
```

Esercizio 3

I seguenti array `prezzo_A`, `prezzo_B` e `prezzo_C` contengono i prezzi in lire di tre titoli azionari nel periodo di 10 giorni. Utilizzare Matlab per trovare:

a) il numero dei giorni in cui il prezzo dell'azione A è stato maggiore sia del prezzo di B sia del prezzo di C.

b) il numero dei giorni in cui il prezzo dell'azione A è stato maggiore del prezzo di B o del prezzo di C.

c) il numero dei giorni in cui il prezzo dell'azione A è stato maggiore del prezzo di B o del prezzo di C, ma non di entrambi.

```
prezzo_A = [190, 180, 220, 210, 250, 190, 170, 210, 270, 290]
prezzo_B = [220, 170, 200, 190, 240, 180, 160, 250, 280, 270]
prezzo_C = [170, 130, 220, 230, 190, 170, 200, 210, 240, 280]
```

Esercizio 4

Se `x = [-3, 0, 0, 2, 5, 8]` e `y = [-5, -2, 0, 3, 4, 10]`, calcolare a mano i risultati delle seguenti operazioni e controllarli con Matlab.

a) `z = y < -x`

b) `z = x&y`

c) `z = x|y`

d) `z = xor(x,y)`

Esercizio 5

Siano e1 ed e2 sono due espressioni logiche. Le leggi di DeMorgan sulle espressioni logiche stabiliscono che:

NOT(e1 AND e2) implica (NOT e1) OR (NOT e2)

e

NOT(e1 OR e2) implica (NOT e1) AND (NOT e2)

Utilizzare queste leggi per trovare un'espressione equivalente per ciascuna delle seguenti espressioni; verificare l'equivalenza con Matlab.

a) `~((x < 10) & (x >= 6))`

b) `~((x == 2) | (x > 5))`

Esercizio 6

Le seguenti espressioni sono equivalenti? Per verificare le risposte, utilizzare Matlab assegnando specifici valori alle variabili *a*, *b*, *c* e *d*.

a) 1. `(a==b) & ((b==c) | (a==c))`
   2. `(a==b) | ((b==c) & (a==c))`

b) 1. `(a<b) & ((a>c) | (a>d))`
   2. `(a<b) & (a>c) | ((a<c)) & (a>d))`

Esercizio 7

Il prezzo in dollari di un determinato titolo azionario nel periodo di 10 giorni è dato del seguente array:

`prezzo = [19, 18, 22, 21, 25, 19, 17, 21, 27, 29]`

Un investitore possiede 1000 azioni all'inizio del periodo di 10 giorni e vuole comprare 100 azioni ogni giorno in cui il prezzo scende sotto i 20 dollari e vendere 100 azioni quando il prezzo supera i 25 dollari. Utilizzare Matlab per calcolare:

a) La spesa totale per acquistare le azioni;

b) L'importo totale derivante dalla vendita delle azioni;

c) Il numero totale di azioni che possiede l'investitore dopo 10 giorni.

d) L'incremento netto del valore del portafoglio azionario.

Per ottenere l'incremento netto del portafoglio azionario è necessario effettuare la **sottrazione** tra

- Prezzo del titolo azionario, al giorno di vendita (giorno 10), moltiplicato per il numero di titoli posseduti (calcolato al punto **c)**)
- Prezzo del titolo azionario, al giorno d'acquisto (giorno 1), moltiplicato per il numero di titoli posseduti (1000)

Esercizio 8

L'altezza e la velocità di un oggetto che viene lanciato con una velocità iniziale v_0 e un angolo A (rispetto al piano orizzontale) sono date dalle seguenti formule:

$$h(t) = v_0 t \sin A - 0{,}5 g t^2$$

$$v(t) = \sqrt{v_0^2 - 2 v_0 g t \sin A + g^2 t^2}$$

Il termine g rappresenta l'accelerazione di gravità. L'oggetto cadrà al suolo quando $h(t) = 0$, nell'istante $t_{suolo} = 2(v_0/g)\sin A$. Siano dati i seguenti valori: $A = 30°$, $v_0 = 40$ m/sec e $g = 9{,}81$ m/sec^2. Utilizzare gli operatori logici e relazionali di Matlab per calcolare gli istanti in cui:

a) L'altezza non è minore di 15 m.

b) L'altezza non è minore di 15 m e contemporaneamente la velocità non è maggiore di 36 m/sec.

c) L'altezza è minore di 5 m o la velocità è maggiore di 35 m/sec.

Esercizio 9

Riscrivere le seguenti espressioni in modo da utilizzare una sola istruzione `if`.

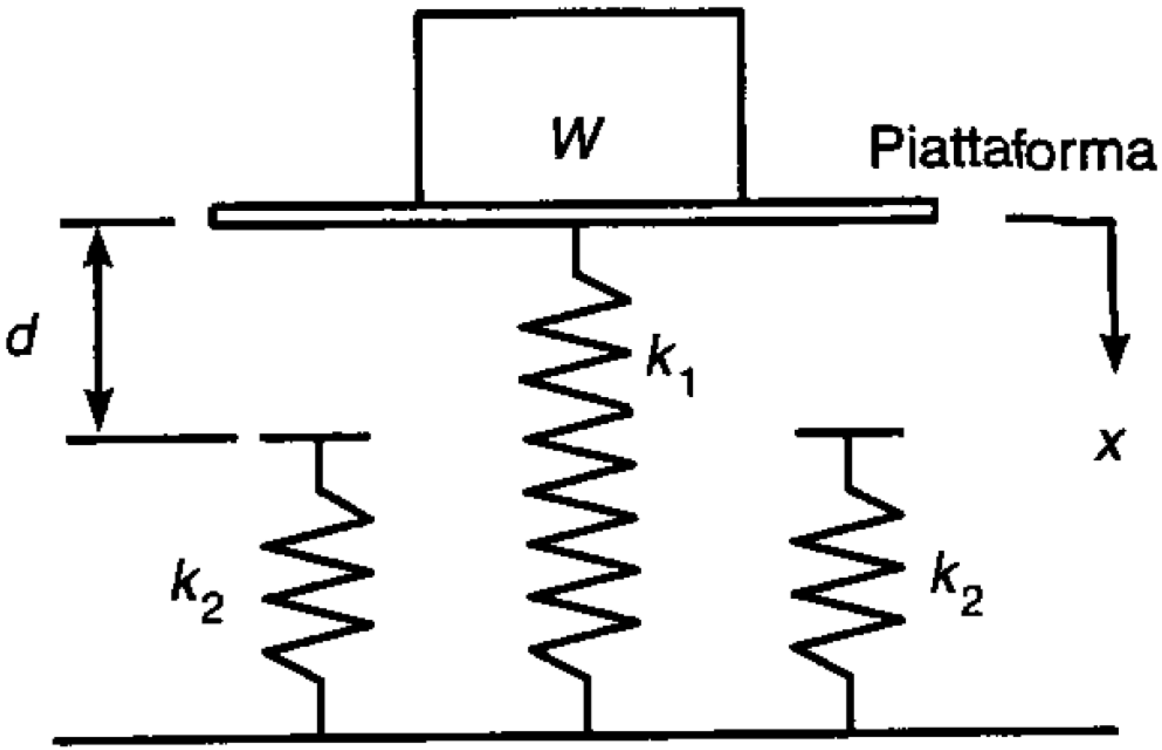
```
if x < y
   if z < 10
      w = x*y*z
   end
end
```

Esercizio 10

- Scrivere una funzione che **<u>prende in input</u>** un parametro *F*, che esprime i gradi in Fahrenheit, e **<u>restituisce in output</u>** i corrispondenti gradi in Celsius

- ***<u>Nota</u>***:
  - **°C = (°F – 32) / 1,8**

Esercizio 11 – 1/3



Esercizio 11 – 2/3

La Figura 4.12 a) illustra un modello massa-molla del tipo utilizzato per progettare le sospensioni dei veicoli. Le molle esercitano una forza che è proporzionale alla loro compressione; il fattore di proporzionalità è la costante elastica k della molla. Le due molle laterali servono a fornire una resistenza aggiuntiva quando il peso W sollecita troppo la molla centrale. Se il peso viene appoggiato sulla piattaforma, il sistema si sposta a una distanza x prima di fermarsi. Affinché il sistema sia in equilibrio statico, la forza peso deve bilanciare le forze delle molle in questa nuova posizione, cioè:

$$W = k_1 x \qquad \text{se } x < d$$

$$W = k_1 x + 2k_2(x - d) \qquad \text{se } x \geq d$$

Esercizio 11 – 3/3

Queste relazioni possono essere utilizzate per generare il diagramma di x in funzione di W, come illustra la Figura 4.12 b).

a) Creare un file di funzione che calcola la distanza x, utilizzando i parametri di input W, k_1, k_2 e d. Provare la funzione per i seguenti due casi, utilizzando i valori $k_1 = 10^4$ N/m; $k_2 = 1{,}5\times10^4$ N/m; $d = 0{,}1$ m.

$W = 500$ newton

$W = 2000$ newton

Esercizio 12 – 1/2

Analizzare il sistema massa-molla descritto nel precedente Problema nel caso in cui il peso W viene lasciato cadere sulla piattaforma attaccata alla molla centrale. Se il peso cade da un'altezza h rispetto alla piattaforma, è possibile calcolare la compressione massima della molla x eguagliando l'energia potenziale di gravità Wh con l'energia potenziale immagazzinata nelle molle:

$$Wh = \frac{1}{2} k_1 x^2 \qquad \text{se } x < d$$

Questa equazione può essere risolta in funzione di x:

$$x = \sqrt{\frac{2Wh}{k_1}} \qquad \text{se } x < d$$

e

$$Wh = \frac{1}{2} k_1 x^2 + \frac{1}{2}(2k_2)(x-d)^2 \qquad \text{se } x \geq d$$

dalla quale è possibile ottenere la seguente equazione di secondo grado in x:

$$(k_1 + 2k_2)x^2 - 4k_2 dx + 2k_2 d_2 - 2Wh = 0 \qquad \text{se } x \geq d$$

Esercizio 12 – 2/2

Creare un file di funzione che calcola la compressione massima x dovuta al peso che cade da un'altezza h. I parametri di input della funzione sono k_1, k_2, d, W e h. Provare la funzione per i seguenti due casi, utilizzando i valori $k_1 = 10^4$ N/m; $k_2 = 1{,}5\times10^4$ N/m e $d = 0{,}1$ m.

$W = 100$ N, $h = 0{,}5$ m

$W = 2000$ N, $h = 0{,}5$ m

Esercizio 13

- Sapendo che un parcheggio richiede **N €** per la **prima ora** di stazionamento e **N-M € per le ore** di stazionamento **successive** alla prima, scrivere una funzione che
- **Prenda in input**
  - Il **numero di ore** che un veicolo ha trascorso in stazionamento
  - I valori di **N** ed **M**
- **Restituisce in output l'importo complessivo** che il proprietario del veicolo deve pagare al parcheggio

- **Esempio**: num_ore = 3, N = 3, M = 1
  - Output funzione = 7 (Importo complessivo da pagare al parcheggio)