ESERCITAZIONE MATLAB 1: **Vettori, matrici ed aritmetica finita**

1. Generare i vettori riga `x` ed `y` contenenti i valori equidistanti $1, 2, \ldots, 10$ e $20, 18, \ldots, 10$, rispettivamente.

2. Generare il vettore colonna `z` contenente 5 valori equidistanti nell'intervallo $\left[0, \frac{1}{2}\right]$.
Suggerimento: si utilizzino il comando `linspace` e l'operatore di trasposizione.

3. Dato il vettore `x` di elementi $-5, -4, \ldots, 8, 9$ determinare l'elemento massimo, minimo, di valore assoluto massimo, di valore assoluto minimo, la somma degli elementi, la somma dei valori assoluti degli elementi.
Nota: il comando Matlab per calcolare il valore assoluto di uno scalare reale è `abs`.

4. Dato il vettore `z=[50:-5:10]` cosa restituiscono i seguenti comandi?

   a) `length(z)`
   b) `z(1:2:9)=zeros(1,5)`
   c) `z(2:2:8)=z(2:2:8)-5*ones(1,4)`
   d) `z([2 8])=z([8 2])`

5. Definire i vettori `x=2:5` e `y=-3:2:3`. Dopo aver eseguito il comando `format rat` che seleziona il formato razionale per la visualizzazione dei numeri, si eseguano le seguenti istruzioni e se ne spieghi il risultato:

   a) `x.*y`
   b) `x./y`
   c) `x.^y`
   d) `x*y`

6. Data la matrice

   `A = [16  10  2; 4  6  14; 8  12  18]`

   esaminare il contenuto di:

   `A(1,1),  A(2,1),  A(1,2),  A(3,3),`
   `A(:,1),  A(3,:),`
   `A(1:2,:),  A(:,2:3),  A([1 3],[1 3]).`

7. Eseguire le seguenti istruzioni e spiegare il risultato:

   a) `A = [1  5  8;  2  0  3]`
   b) `x = [-1  10  1]`
   c) `y = [6;  7;  9]`
   d) `size(A)`
   e) `B = [A; x]`
   f) `B(:,3) = y`

8. Dati la matrice `A = [1  3;  4  -2]` ed il vettore `b = [2; -1]`

   calcolare:

   `A+A,  A-A,  A*A,  A^2,  A.*A,  A.^2,  A*b,   A'*b,  b'*A.`

9. Data la matrice 5×5

   `A=[1:5; 6:10; 11:15; 16:20; 21:25]`

   a) generare la matrice `U` triangolare superiore i cui elementi diversi da zero coincidono con gli omonimi di A;
   b) generare la matrice `L` triangolare inferiore i cui elementi diagonali valgono 1 mentre gli elementi della porzione strettamente triangolare inferiore, ovvero appartenenti alle diagonali di indice $k < 0$, coincidono con quelli omonimi di A;
   c) generare le matrici `T`, `BS` e `BI` rispettivamente tridiagonale, bidiagonale superiore e bidiagonale inferiore, i cui elementi non nulli coincidono con gli omonimi di A;

   Suggerimento: Utilizzare i comandi `tril, triu` e `eye`.

10. Applicare i seguenti due algoritmi per il calcolo della radice positiva della seguente equazione di secondo grado

$$y^2 + 2py - q = 0$$

   con $p = 4.\underbrace{99999999999}_{11 \text{ volte}}5 \cdot 10^4$ e $q = 10^{-2}$.

   Algoritmo 1:

```
format long
delta = p^2+q
r = sqrt(delta)
y1 = -p + r
```

   Algoritmo 2:

```
format long
delta = p^2+q
r  = sqrt(delta)
y2  = -p - r
y1 = -q/y2
```

   La soluzione esatta è $y_1 = 10^{-7}$. Spiegare il motivo per cui il primo algoritmo fornisce un risultato inaccurato.

11. Spiegare il risultato che si ottiene digitando il seguente comando
   `(1+eps/4)>1`

12. Determinare il più efficace algoritmo per calcolare, in aritmetica finita,

$$\sqrt{x^2 + y^2}.$$

   Suggerimento: l'algoritmo deve prevenire per quanto possibile l'overflow.
   Si testi la procedura prendendo $x = 2^{600}$ e $y = x\,\sqrt{u/2}$ dove u denota la precisione di macchina (ovvero `u=eps/2` in Matlab).

SOLUZIONI:

1. `x=1:10;`
   `y=20:-2:10;`

2. `z=linspace(0,0.5,5)';`

3. `x = -5:9;`

`max(x)`	(`ans` $=$ **9**)	`min(x)`	(`ans` $= -$**5**)	`max(abs(x))`	(`ans` $=$ **9**)
`min(abs(x))`	(`ans` $=$ **0**)	`sum(x)`	(`ans` $=$ **30**)	`sum(abs(x))`	(`ans` $=$ **60**)

4. a) `length(z)` restituisce la lunghezza del vettore `z`, cioè **9**.

   b) `z([1:2:9])=zeros(1,5)` modifica le componenti di `z` di indice dispari, ovvero di indice $1,3,5,7,9$, assegnando il valore 0 ad ognuna di esse. Si sarebbe ottenuto lo stesso risultato anche digitando `z([1:2:9])=0;`

   c) `z([2:2:8])=z([2:2:8])-5*ones(1,4)` aggiorna le componenti di indice pari di `z` sottraendo `5` ad ognuna di esse. Si sarebbe ottenuto lo stesso risultato anche digitando `z([2:2:8])=z([2:2:8])-5;`

   d) `z([2 8])=z([8 2])` scambia il valore della seconda e dell'ottava componente di `z`.

5. a) `x.*y` restituisce il vettore `ans` contenente il prodotto componente per componente tra i vettori `x` e `y` (`ans = [-6 -3 4 15]`);

   b) `x./y` restituisce il vettore `ans` contenente il rapporto componente per componente tra i vettori `x` e `y` (`ans = [-2/3 -3 4 5/3]`);

   c) `x.^y` restituisce il vettore `ans` contenente le componenti di `x` elevate alla potenza specificata dalle corrispondenti componenti di `y` (`ans = [1/8 1/3 4 125]`);

   d) lanciando l'esecuzione di `x*y` si chiede di calcolare il prodotto righe per colonne tra due vettori riga. Tale operazione non è definita e pertanto Matlab genera un messaggio di errore.

6. `A(1,1) = 16,` `A(2,1) = 4,` `A(1,2) = 10,` `A(3,3) = 18,`
   `A(:,1)` restituisce la prima colonna di `A`,
   `A(3,:)` restituisce la terza riga di `A`,
   `A(1:2,:)` restituisce la sottomatrice di `A` costituita dalle sue prime due righe e da tutte le sue colonne ovvero $\begin{pmatrix} 16 & 10 & 2 \\ 4 & 6 & 14 \end{pmatrix}$,
   `A(:,2:3)` restituisce la sottomatrice di `A` costituita da tutte le sue righe e dalla seconda e terza colonna ovvero $\begin{pmatrix} 10 & 2 \\ 6 & 14 \\ 12 & 18 \end{pmatrix}$
   `A([1 3],[1 3])` restituisce la sottomatrice di `A` costituita dalla sua prima e terza riga e dalla sua prima e terza colonna ovvero $\begin{pmatrix} 16 & 2 \\ 8 & 18 \end{pmatrix}$

7.

   a,b,c) Le prime tre istruzioni definiscono una matrice `A` di dimensione 2×3, un vettore riga `x` di lunghezza 3 ed un vettore colonna `y` della stessa lunghezza, rispettivamente;

d) `size(A)` restituisce la dimensione della matrice `A`. Più nel dettaglio `size(A)` restituisce un vettore di lunghezza 2 la cui prima componente è il numero di righe di `A` (cioè 2) mentre la seconda componente è il numero di colonne di `A` (ovvero 3);

e) `B=[A;x]` definisce la matrice `B` di dimensione 3×3 le cui prime due righe coincidono con quelle di `A` mentre la terza coincide con il vettore riga `x`;

f) `B(:,3) = y` sostituisce i valori della terza colonna di `B` con gli omonimi del vettore colonna `y`.

8. `A+A`$= \begin{pmatrix} 2 & 6 \\ 8 & -4 \end{pmatrix}$ `A-A`$= \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$ `A*A`$=$`A^2`$= \begin{pmatrix} 13 & -3 \\ -4 & 16 \end{pmatrix}$

   `A.*A`$=$`A.^2`$= \begin{pmatrix} 1 & 9 \\ 16 & 4 \end{pmatrix}$

   `A*b`$= \begin{pmatrix} -1 \\ 10 \end{pmatrix}$ `A'*b`$= \begin{pmatrix} -2 \\ 8 \end{pmatrix}$ `b'*A`$= \begin{pmatrix} -2 & 8 \end{pmatrix} =$`(A'*b)'`

9. a) `U` $=$ `triu(A)`;

   b) `L` $=$ `tril(A,-1)`+eye(5);

   c) `T= triu(tril(A,1),-1);   BS=triu(T);   BI=tril(T);`

10. Il primo algoritmo fornisce un risultato inaccurato poichè $p^2 \gg q$ e quindi $r = \sqrt{p^2+q} \approx p$. Questo implica che la somma algebrica $-p+r$, richiesta dalla terza istruzione del primo algoritmo, è mal condizionata ed, in particolare, si verifica il fenomeno della cancellazione numerica.
    Le operazioni coinvolte nel secondo algoritmo, invece, sono tutte ben condizionate e quindi il risultato finale è accurato compatibilmente con la precisione di macchina dello standard IEEE per la doppia precisione.

11. Digitando il comando indicato si ottiene `0` da interpretarsi come `falso`. Considerando infatti che `eps/4` è strettamente compreso tra 0 e la precisione di macchina `u=eps/2` si ha che, in aritmetica finita, `1+eps/4=1` e quindi non è strettamente maggiore di 1.

12. Se x oppure y sono molto grandi, in valore assoluto, allora il calcolo di x^2 o y^2 può provocare overflow. Al fine di prevenire per quanto possibile tale problema, quando $|x| \geq |y|$, conviene calcolare $\sqrt{x^2+y^2}$ usando la seguente espressione equivalente

$$|x|\sqrt{1+(y/x)^2}\,.$$

In questo modo, infatti, $|y/x| < 1$ e quindi il calcolo di $(y/x)^2$ non provoca sicuramente overflow.
In modo analogo, l'espressione che conviene usare quando $|y| \geq |x|$ è

$$|y|\sqrt{1+(x/y)^2}\,.$$