



Functions

Loris Giulivi – Nicolò Folloni

Come definire una function

Le funzioni sono delle parti di codice che permettono di essere richiamate all'interno di uno script. Sono particolarmente utili per non dover riscrivere più volte la stessa istruzione all'interno del codice.

In Matlab, le funzioni sono identificate dalla parola chiave `function`, e sono strutturate nel seguente modo:

```
function [output1,output2] = NomeFunction(input1,input2)
% Descrizione della funzione: specificare quali sono input e quali gli
% output e cosa calcola la function

% istruzioni e definizione degli output:
output1 = input1;
output2 = input2;

end
```

Function

- Se una function ha M output ma siamo interessati solo ai primi N, basta richiedere in output solo N valori
- Se siamo interessati al secondo output di una function ma non al primo, basta sostituire il primo output con ~

```
[somma, differenza] = Operazioni(10,5);
```

```
[~, ~, prodotto] = Operazioni(10,5);
```

```
function [somma, differenza, prodotto] = Operazioni(x,y)
% INPUT: due numeri reali x, y.
% OUTPU: somma, differenza e prodotto dei numeri x, y.

somma = x+y;
differenza = x-y;
prodotto = x*y;

end
```

Esempio

Data la function ValoreAssoluto, cosa stampa la seguente riga di codice?

```
[~, V] = ValoreAssoluto(-18);
```

```
function [output1, output2] = ValoreAssoluto(input)

    if input >= 0
        output1 = input;
        output2 = input;
    else
        output1 = -input;
        output2 = input;
    end
end
```

Le funzioni più comuni

zeros(m, n): crea una matrice di zeri di dimensione mxn

ones(m, n): crea una matrice di uni di dimensione mxn

eye(n): crea la matrice identità di dimensione nxn

rand(m, n): crea una matrice random con numeri compresi tra 0 e 1 di dimensione mxn

randi(p,m,n): crea una matrice random con numeri compresi tra 1 e p di dimensione mxn

length(x): restituisce la dimensione del vettore x

size(A): restituisce le dimensioni della matrice A

max(v): restituisce il massimo del vettore v e il suo indice

min(v): restituisce il minimo del vettore v e il suo indice

mean(v): restituisce la media del vettore v

mod(n,p): restituisce il resto della divisione intera tra n e p

find(p): restituisce gli indici degli elementi che soddisfano p

Per avere informazioni complete su una funzione, usare il comando *help*

```
>> help max
max      Maximum elements of an array.
M = max(X) is the largest element in the vector X. If X is a matrix, M
is a row vector containing the maximum element from each column. For
N-D arrays, max(X) operates along the first non-singleton dimension.

When X is complex, the maximum is computed using the magnitude
max(ABS(X)). In the case of equal magnitude elements the phase angle
max(ANGLE(X)) is used.

[M,I] = max(X) also returns the indices into operating dimension
corresponding to the maximum values. If X contains more than one
element with the maximum value, then the index of the first one
is returned.

C = max(X,Y) returns an array with the largest elements taken from X or
Y. X and Y must have compatible sizes. In the simplest cases, they can
be the same size or one can be a scalar. Two inputs have compatible
sizes if, for every dimension, the dimension sizes of the inputs are
either the same or one of them is 1.

M = max(X,[],'all') returns the largest element of X.

M = max(X,[],DIM) or [M,I] = max(X,[],DIM) operates along the
dimension DIM.

M = max(X,[],VECDIM) operates on the dimensions specified in the vector
VECDIM. For example, max(X,[],[1 2]) operates on the elements contained
in the first and second dimensions of X.
```

Esercizio 1

ES1: Scrivere una function che dato il raggio di una sfera ne calcola il volume.
Scrivere uno script per determinare qual è il minimo intero positivo R tale che la sfera di raggio R abbia volume minore di R^4 .

Esercizio 1 - Soluzione

```
R = 1;
flag = 0;
while flag == 0
    V = volumeSfera(R);
    if V < R^4
        flag = 1;
    else
        R = R + 1;
    end
end
```

```
txt = ['Il numero cercato è R = ', num2str(R)];
disp(txt);
```

```
function [V] = volumeSfera(R)
V = (4/3)*pi*R^3;
end
```

Esercizio 2

ES2: Scrivere una function che presi in input Q punti in uno spazio euclideo a dimensione generica N restituisce come output una matrice D con le distanze euclidee tra ciascun punto. $D(i,j)$ quindi conterrà la distanza tra il punto i e il punto j .

Testare la function con uno script che chiedi in input all'utente N e Q stabilisce quale coppia di punti distinti ha distanza massima, quale minima, e quante coppie di punti hanno distanza maggiore di π greco.

Esercizio 2 - Soluzione

```
clc
clear

N = input('Inserisci dimensione spazio N: ');
Q = input('Inserisci numero di punti Q: ');

Punti = rand(N,Q)*100;
D = distanze(Punti);
```

```
max_val = -inf;
min_val = inf;
counter = 0;
```

```
% METODO ALLA C
```

```
for ii = 1:Q
    for jj = ii+1:Q
        if D(ii,jj) > max_val
            max_val = D(ii,jj);
            ind_max = [ii, jj];
        elseif D(ii,jj) < min_val
            min_val = D(ii,jj);
            ind_min = [ii,jj];
        end

        if D(ii,jj) > pi
            counter = counter + 1;
        end
    end
end
```

```
% METODO ALLA MATLAB
```

```
[M_max, I_max] = max(D);
[~, i_max] = max(M_max);
```

```
t = zeros(Q,1) + inf;
Dmin = D + diag(t);
```

```
[M_min, I_min] = min(Dmin);
[~, i_min] = min(M_min);
```

```
max_pi = sum((D>pi), 'all')/2;
```

```
% VISUALIZZAZIONE SOLUZIONI
```

```
txt1 = ['I punti in posizione ', num2str(ind_max(1)), ' e ', num2str(ind_max(2)), ' hanno distanza massima'];
txt2 = ['I punti in posizione ', num2str(ind_min(1)), ' e ', num2str(ind_min(2)), ' hanno distanza minima'];
txt3 = ['Ci sono ', num2str(counter), ' coppie di punti con distanza maggiore di pi greco'];
disp(txt1);
disp(txt2);
disp(txt3);
```

```
function [D] = distanza(Punti)
```

```
[~, Q] = size(Punti);
D = zeros(Q,Q);
```

```
for ii = 1:Q
```

```
    for jj = ii+1:Q
        D(ii, jj) = sqrt(sum((Punti(:,ii)-Punti(:,jj)).^2));
```

```
    end
```

```
end
```

```
D = D + D';
```

```
end
```

Esercizio 3

ES3: Ad un esame di guida sono presenti due esaminatori, Marco e Anna, che valutano i candidati secondo criteri leggermente diversi. La matrice punteggiata rappresenta un candidato per ogni riga, sulla prima colonna contiene il numero di errori non gravi commessi da ogni candidato durante l'esame, sulla seconda il numero di errori gravi. Marco bocchia un candidato se almeno una delle seguenti situazioni si verifica: (i) il candidato ha commesso due o più errori gravi, (ii) il candidato ha commesso uno o più errori gravi e due o più errori non gravi, (iii) il candidato ha commesso quattro o più errori non gravi. Anna bocchia un candidato se si verifica almeno una delle seguenti situazioni: (i) il candidato ha commesso uno o più errori gravi, (ii) il candidato ha commesso cinque o più errori non gravi. Marco inizia ad esaminare i candidati dal fondo della lista, mentre Anna dall'inizio.

Scrivere due function, una per Marco e una per Anna, che preso in ingresso il numero di errori commessi, stabiliscono se il candidato in questione è bocciato o promosso.

Scrivere uno script che simuli l'esame di guida, stampando a video il nome del candidato e se ha passato o meno l'esame. Stabilire poi se tra i candidati ce n'è qualcuno il cui esito sarebbe stato diverso se esaminato dall'altro esaminatore.