

11 Funzioni MATLAB

Le funzioni in un programma sono utilizzate per strutturare il codice in sottoparti e per evitare di replicare inutilmente il codice. In MATLAB le funzioni vengono identificate con la parola chiave `function` ed è buona regola che abbiano lo stesso nome dello script che le contiene. In generale sono strutturate nel seguente modo:

```
1 function [output1, output2, ..] = nome(input1, input2, ..)
2 %istruzioni
3 output1 = ..
4 output2 = ..
5 ...
```

dove

- `output1, output2, ..` sono gli output (opzionali), che, se dichiarati, devono essere inizializzati dalla funzione
- `input1, input2, ..` sono gli input (opzionali) che servono per il calcolo degli output

Prima di iniziare a scrivere il corpo di una funzione, la prima cosa da stabilire è quali siano input (argomenti) e output (valori restituiti) necessari.

MATLAB vede solo le funzioni delle proprie librerie (come `mean()`, `min()`) o le funzioni dichiarate nella stessa cartella di dove viene eseguito lo script in cui si richiamano le funzioni.

Un elenco (non esaustivo) delle funzioni di MATLAB è il seguente:

```
1 zeros(m,n) %crea una matrice di zeri di dimensioni m*n
2 ones(m,n) %crea una matrice di uni di dimensioni m*n
3 eye(n) %crea una matrice identita' di ordine n
4 rand(m,n) %crea una matrice di numeri casuali in [0,1] di
   dimensioni m*n
5
6 length(v) %restituisce la lunghezza del vettore v
7 size(M) %restituisce le dimensioni della matrice M
8
9 ceil(x) % arrotonda x all'intero superiore
```

```
10 floor(x) %arrotonda x all'intero inferiore
11 fix(x) %arrotonda x all'intero piu' vicino a 0
12
13 max(v) %restituisce il massimo del vettore v
14 min(v) %restituisce il minimo del vettore v
15 mean(v) %restituisce la media del vettore v
16 mod(m,n) % restituisce il modulo n di m
17
18 find(p) %restituisce gli indici degli elementi che soddisfano p
```

11.1 Esercizi

Esercizio 11.1

Scrivere uno script che dato un numero intero positivo n minore di 6765, stabilisca se è di Fibonacci e nel caso non lo sia restituisca il più grande numero di Fibonacci minore di n e il più piccolo numero di Fibonacci maggiore di n .

Strutturare lo script in modo da utilizzare una funzione che dato un numero $N = 20$ restituisca i primi N numero di Fibonacci.

Esercizio 11.2

Scrivere una funzione `closestVal` che prende in ingresso:

- un vettore v
- un valore n

e restituisce un valore di v più vicino a n , ossia a distanza minima.

Ad esempio:

```
1 v = [1 4 40];  
2 n = 20;  
3 closestVal(v, n)
```

restituisce 4, mentre

```
1 v = [1 4 40];  
2 n = 32;  
3 closestVal(v, n)
```

restituisce 40.

Commentare il caso in cui si debbano restituire tutti i valori a distanz minim (nel caso ce ne sia più di uno).

Esercizio 11.3

Scrivere uno script che chiede all'utente di inserire un numero positivo a (nel caso in cui il numero non è positivo ripetere l'inserimento) e verifichi se il numero è perfetto, in caso contrario dice se è abbondante o difettivo.

Un numero è perfetto se corrisponde alla somma dei suoi divisori, escluso se stesso (6 è perfetto $1 + 2 + 3 = 6$), è abbondante se è minore della somma dei suoi divisori (20 visto che $1 + 2 + 4 + 5 + 10 > 20$) e altrimenti è difettivo (19 e tutti i numeri primi sono difettivi).

Richiede un altro numero b e controlla se i due numeri sono amici due numeri a, b

Due numeri a e b sono amici se la somma dei divisori di a è uguale a b e viceversa (es 284 e 220 sono amici).

Strutturare lo script con delle funzioni:

- `inserisciPositivo` che legge un numero intero positivo;
- `sommaDivisori` che dato un numero restituisce la somma dei suoi divisori;
- `controllaSePerfetto` che restituisce se un numero è perfetto, se è abbondante o difettivo:
- `controllaSeAmici` che restituisce se due numeri sono amici.

Esercizio 11.4

Quest'anno Babbo Natale ha deciso di farsi aiutare per la consegna dei regali da KwanzaBot e Superman.¹ Babbo natale ha un elenco di bambini (`elenco` matrice di caratteri) e deve consegnare un regalo particolare ad ognuno di essi tra piccolo (1), medio (2) e grande (3) a seconda di quanto sono stati buoni (`buono` vettore di interi). Per assegnare il regalo ci sono delle soglie di punteggi: fino a 700 punti il bambino ha un regalo piccolo, fino a 900 medio, altrimenti grande. Ad ogni consegna si deve stampare a video il nome del bambino, il suo regalo e chi l'ha portato.

Per dividersi il lavoro i tre hanno deciso che Babbo Natale avrebbe consegnato i regali partendo dal primo dell'elenco, KwanzaBot dal fondo e Superman scegliendo uno dei bambini a caso (`randi()`). I bambini che hanno il regalo grande lasciano anche dei biscotti a chi consegna il regalo, quindi ringrazia riportando il suo apprezzamento per i biscotti (stampandolo a video).

Strutturare lo script che consegni i regali a tutti i bambini in funzioni che dividano in maniera logica l'azione di selezione del regalo (`selezionaRegalo`), la consegna del regalo (`consegnaRegalo`) e l'eliminazione dall'elenco di chi ha già ricevuto il regalo (`cancellaBambino`).

¹<http://futurama.wikia.com/wiki/Kwanzaabot>

Soluzioni

Soluzione dell'esercizio 11.1

```
1 clear
2 clc
3 close all
4
5 N = 20;
6 F = fibonacci(N);
7
8 n = input(' Inserire un numero ');
9
10 if(any(F == n))
11     disp([num2str(n), ' e' di Fibonacci'])
12 else
13     indx_M = find(F > n);
14     M = F(indx_M(1));
15
16     indx_m = find(F < n);
17     m = F(indx_m(end));
18
19     str = [num2str(n), ' e' compreso tra '];
20     str = [str, num2str(m), ' e ', num2str(M)];
21     disp(str);
22 end
```

```
1 function F = fibonacci(n)
2 % FIBONACCI calcola i primi n numeri di fibonacci
3
4 F = zeros(n,1);
5
6 if(n >= 1)
7     F(1) = 1;
8 end
9 if (n >= 2)
10    F(2) = 1;
11 end
12
13 for ii = 3 : n
14    F(ii) = F(ii-1) + F(ii-2);
15 end
```

Soluzione dell'esercizio 11.2

```
1 clc
2 clear
3 close all
4
5 %% Sezione 1
6 v = [1 4 40];
7 n = 20;
8 closestVal(v, n)
9
10 n = 32;
11 closestVal(v, n)
12
13 %% Sezione 2
14 vett = [10 3 -5 7 5];
15 valore = 4;
16
17 res = closestVal(vett, valore);
```

```
1 function [val] = closestVal(v, n)
2 % CLOSESTVAL prende in ingresso un vettore v ed un valore n e
3 % restituisce il valore di v più vicino a n
4
5 dist = abs(v - n);
6 [~, pos] = min(dist);
7 val = v(pos);
```

```
1 function [val] = closestVal2(v, n)
2 % CLOSESTVAL prende in ingresso un vettore v ed un valore n e
3 % restituisce i valori di v più vicini a n
4
5 dist = abs(v - n);
6 pos = dist == min(dist);
7 val = v(pos);
```

Soluzione dell'esercizio 11.3

```
1 clc
2 clear
3
4 % richiedere numero
5 n = inserisciPositivo();
```

```
6
7 [perf, abb, dif] = controllaSePerfetto(n);
8
9 if(perf == 1)
10     disp([num2str(n), ' e'' perfetto']);
11 else
12     disp([num2str(n), ' NON e'' perfetto']);
13     if(abb == 1)
14         disp([num2str(n), ' e'' abbondante']);
15     else
16         disp([num2str(n), ' e'' difettivo']);
17     end
18 end
19
20 m = inserisciPositivo();
21
22 amici = controllaSeAmici(n,m);
23
24 if(amici)
25     disp([num2str(n), ' e ', num2str(m), ' sono amici'])
26 else
27     disp([num2str(n), ' e ', num2str(m), ' NON sono amici'])
28 end
```

```
1 function n = inserisciPositivo()
2
3 n = -1;
4 while(n < 0)
5     n = input('Inserire un numero positivo ');
6 end
```

```
1 function s = sommaDivisori(n)
2
3 D = 1 : 1 : n / 2;
4 A = mod(n, D);
5 s = sum(D(A == 0));
```

```
1 function s = sommaDivisoriCLike(n)
2
3 v = [];
4 for ii = 1 : n / 2
5     if(mod(n, ii) == 0)
6         v = [v, ii];
```

```
7     end
8 end
9 s = sum(v);
```

```
1 function [perf, abb, dif] = controllaSePerfetto(n)
2
3 if(n == sommaDivisori(n))
4     perf = 1;
5     abb = 0;
6     dif = 0;
7 elseif(n < sommaDivisori(n))
8     perf = 0;
9     abb = 1;
10    dif = 0;
11 else
12    perf = 0;
13    abb = 0;
14    dif = 1;
15 end
```

```
1 function res = controllaSeAmici(n, m)
2
3 if (sommaDivisori(n) == m && sommaDivisori(m) == n)
4     res = 1;
5 else
6     res = 0;
7 end
```

Soluzione dell'esercizio 11.4

```
1 clear
2 clc
3 close all
4
5 %%
6 load('dati_babbo_natale');
7 n_rimasti = size(elenco,1);
8
9 while n_rimasti > 0
10
11     %BN
12     regaloBN = selezionaRegalo(buono,1);
```

```

13     elenco = consegnaRegalo(elenco, 1, regaloBN, 'Babbo Natale'
14         );
15     [elenco, buono] = cancellaBambino(elenco, buono, 1);
16     n_rimasti = n_rimasti - 1;
17     %QB
18     if n_rimasti > 0
19         regaloQB = selezionaRegalo(buono, n_rimasti);
20         consegnaRegalo(elenco, n_rimasti, regaloQB, 'QwanzaBot'
21             );
22         [elenco, buono] = cancellaBambino(elenco, buono,
23             n_rimasti);
24         n_rimasti = n_rimasti - 1;
25     end
26     %SM
27     if n_rimasti > 0
28         ind = randi(n_rimasti);
29         regaloSM = selezionaRegalo(buono, ind);
30         consegnaRegalo(elenco, ind, regaloSM, 'Superman');
31         [elenco, buono] = cancellaBambino(elenco, buono, ind);
32         n_rimasti = n_rimasti - 1;
33     end
34     pause();
35 end

```

```

1 function regalo = selezionaRegalo(buono, ind)
2
3
4 if buono(ind) < 700
5     regalo = 1; %regalo piccolo
6 elseif buono(ind) < 900
7     regalo = 2; %regalo medio
8 else
9     regalo = 3; %regalo grande
10 end

```

```

1 function consegnaRegalo(elenco, idx, regalo, identita)
2
3 fprintf(['Quest''anno ' elenco(idx,:) ' ha ricevuto un regalo
4     di tipo ']);
5 if regalo == 1
6     fprintf('piccolo');
7 elseif regalo == 2

```

```
7     fprintf('medio');
8 else
9     fprintf('grande\n');
10    fprintf([identita ' ha gradito i biscotti']);
11 end
12 fprintf('\n');
```

```
1 function [elenco, buono] = cancellaBambino(elenco, buono, ind)
2
3 buono(ind) = [];
4 elenco(ind,:) = [];
```