

 POLITECNICO DI MILANO

Dipartimento di
Elettronica e Informazione

Ottava esercitazione

Riccardo Cattaneo

Alessandro A. Nacci



POLITECNICO
DI MILANO

10/01/2014



Ottava esercitazione: agenda

- (55') Blocco 1: funzioni MATLAB
- (5') Pausa + domande
- (50') Blocco 2: funzioni MATLAB (livello-esame)
- (10') Pausa + domande
- (10') Question time
- (30') Blocco 3: funzioni ricorsive

Blocco 1 / 1 (15')

```
% Si vuole compilare la pagella dei propri film preferiti.  
% 1. Scrivere un programma che chieda di inserire i film. Ogni  
% film e' caratterizzato da un anno, un titolo e un voto.  
% 2. Scrivere il codice che permetta di visualizzare il numero  
% totale di film con voto superiore a 6.  
% 3. Scrivere il codice che permetta di visualizzare i titoli e i  
% voti dei film prodotti tra il 2000 e il 2005.  
  
% Scrivete il codice "matlab-style" _e_ "C-style": cosa cambia?  
% Laddove lo riteniate utile, introducete funzioni ausiliarie  
%      - Quali avete scelto, e perché?
```

Blocco 1 / 1 (15')

CODICE 1

Blocco 1 / 2 / 1 (10')

```
% Scrivere una funzione controllaVettore che riceva in ingresso un
% solo parametro, rappresentato da un vettore (riga o colonna), e
% restituisca un valore logico (true o false)
%
% La funzione deve eseguire queste operazioni:
% 1) Controllare la dimensione del vettore, e restituire false
%     se il parametro in ingresso non è un vettore
% 2) Se il parametro in ingresso invece è un vettore, restituire
%     true se contiene soltanto valori compresi tra 1 e 5, false
%     altrimenti
```

Blocco 1 / 1 (10')

CODICE 2

Blocco 1 / 2 / 2 (15')

```
% Scrivere una funzione controllaMatrici con:
% Due matrici A e B in ingresso
% Due valori r e c in uscita
%
% La funzione deve eseguire le seguenti operazioni:
%
% 1) Verificare che le due matrici abbiano le stesse dimensioni;
%     in caso contrario, restituire r=-1 e c=-1
% 2) Calcolare il valore r come il numero di righe che in entrambe le
%     matrici contengono soltanto numeri compresi tra 1 e 5
% 3) Calcolare il valore c come il numero di colonne che in entrambe le
%     matrici contengono soltanto numeri compresi tra 1 e 5

% Per eseguire i punti 2 e 3 si richiede di utilizzare la funzione
% controllaVettore definita nell'esercizio precedente

% Esempio:
%      M      N
%  1  2  2  1   2  1  3  4   r = 1 (la riga 1 soddisfa la condizione)
%  5  7  2  8   1  5  4  4   c = 2 (le colonne 1 e 3 soddisfano la condizione)
%  3  3  3  4   1  6  1  2
```

Blocco 1 / 2 / 2 (15')

CODICE 3

Blocco 1 / 3 (15')

```
% Scrivere un programma che chieda all'utente di inserire un
numero positivo (in caso contrario ripete la richiesta),

% 1. verifica se il numero e' perfetto.
% 2. Nel caso non sia perfetto, acquisisce un nuovo numero
% positivo e verifica se i due numeri acquisiti sono amici.
%
% Un numero si dice perfetto quando e' uguale alla somma di tutti
% i suoi divisori escluso se' stesso.
%
% In matematica, sono numeri amicabili o amici quelli per cui la
% somma dei divisori propri di uno e' uguale all'altro e
% viceversa.

% faccio 4 funzioni
% function n=inserisciNumero()
% function res=controllaSePerfetto(n)
% function res=controllaSeAmici(n, m)
% function somma=sommaDivisori(n)
```

Blocco 1 / 3 (15')

CODICE 4

Ottava esercitazione: agenda

- (55') Blocco 1: funzioni MATLAB
- (5') Pausa + domande
- (50') Blocco 2: funzioni MATLAB (livello-esame)
- (10') Pausa + domande
- (10') Question time
- (30') Blocco 3: funzioni ricorsive

Ottava esercitazione: agenda

- (55') Blocco 1: funzioni MATLAB
- (5') Pausa + domande
- (50') Blocco 2: funzioni MATLAB (livello-esame)
- (10') Pausa + domande
- (10') Question time
- (30') Blocco 3: funzioni ricorsive

Blocco 2 / 1 / 1 (20')

Una matrice Matlab contiene numeri interi. Si vuole progettare una funzione che ricevendo la matrice e un array di numeri interi che rappresenta una sequenza, cerchi tale sequenza all'interno della matrice. La sequenza può essere disposta, nella matrice, verticalmente dall'alto verso il basso od orizzontalmente, da sinistra verso destra. La funzione deve avere la seguente intestazione:

```
function [riga, col, dir] = cercaInizioSequenza(matrice, seq)
```

se la sequenza è presente nella matrice allora *riga* e *col* indicano gli indici di riga e di colonna del suo primo elemento, mentre *dir* viene posto uguale al carattere 'v' se la sequenza è disposta verticalmente, 'o' se orizzontalmente (se la sequenza è presente ripetuta in più posizioni, i valori restituiti possono essere quelli relativi a una qualsiasi delle ripetizioni); se la sequenza non è presente, *riga* e *col* valgono entrambi 0 e *dir* vale 'n'.

1. Per codificare la funzione in questione, si sviluppino prima le due seguenti funzioni ausiliarie

function [pres] = verificaSeqOrizzontaleDaPosizione (matrice, seq, riga, col)

function [pres] = verificaSeqVerticaleDaPosizione (matrice, seq, riga, col)

che ricercano la sequenza nella matrice a partire da una posizione d'inizio precisa, in direzione orizzontale o verticale: *riga* e *col* sono il punto di inizio; il risultato *pres* vale 1 se la sequenza è presente, 0 altrimenti.

2. Successivamente si utilizzano queste due funzioni per codificare le due seguenti

function [riga, col] cercaInizioSeqOrizzontale = (matrice, seq)

function [riga, col] cercaInizioSeqVerticale = (matrice, seq)

che ricercano la sequenza in tutta la matrice con disposizione orizzontale e verticale, restituendo in *riga* e *col* le coordinate del punto d'inizio, se la sequenza viene trovata, o il valore 0 altrimenti.

3. Infine si codifica la funzione *cercaInizioSequenza* facendo uso delle due precedenti.

Blocco 2 / 1 / 1 (20')

Una matrice Matlab contiene numeri interi. Si vuole progettare una funzione che ricevendo la matrice e un array di numeri interi che rappresenta una sequenza, cerchi tale sequenza all'interno della matrice. La sequenza può essere disposta, nella matrice, verticalmente dall'alto verso il basso od orizzontalmente, da sinistra verso destra. La funzione deve avere la seguente intestazione:

```
function [riga, col, dir] = cercaInizioSequenza(matrice, seq)
```

se la sequenza è presente nella matrice allora *riga* e *col* indicano gli indici di riga e di colonna del suo primo elemento, mentre *dir* viene posto uguale al carattere 'v' se la sequenza è disposta verticalmente, 'o' se orizzontalmente (se la sequenza è presente ripetuta in più posizioni, i valori restituiti possono essere quelli relativi a una qualsiasi delle ripetizioni); se la sequenza non è presente, *riga* e *col* valgono entrambi 0 e *dir* vale 'n'.

- 1. Per codificare la funzione in questione, si sviluppino prima le due seguenti funzioni ausiliarie**
function [pres] = verificaSeqOrizzontaleDaPosizione (matrice, seq, riga, col)
function [pres] = verificaSeqVerticaleDaPosizione (matrice, seq, riga, col)
che ricercano la sequenza nella matrice a partire da una posizione d'inizio precisa, in direzione orizzontale o verticale: *riga* e *col* sono il punto di inizio; il risultato *pres* vale 1 se la sequenza è presente, 0 altrimenti.

2. Successivamente si utilizzano queste due funzioni per codificare le due seguenti
function [riga, col] cercaInizioSeqOrizzontale = (matrice, seq)
function [riga, col] cercaInizioSeqVerticale = (matrice, seq)
che ricercano la sequenza in tutta la matrice con disposizione orizzontale e verticale, restituendo in *riga* e *col* le coordinate del punto d'inizio, se la sequenza viene trovata, o il valore 0 altrimenti.
3. Infine si codifica la funzione *cercaInizioSequenza* facendo uso delle due precedenti.

20'

Blocco 2 / 1 / 1 (20')

Una matrice Matlab contiene numeri interi. Si vuole progettare una funzione che ricevendo la matrice e un array di numeri interi che rappresenta una sequenza, cerchi tale sequenza all'interno della matrice. La sequenza può essere disposta, nella matrice, verticalmente dall'alto verso il basso od orizzontalmente, da sinistra verso destra. La funzione deve avere la seguente intestazione:

```
function [riga, col, dir] = cercaInizioSequenza(matrice, seq)
```

se la sequenza è presente nella matrice allora *riga* e *col* indicano gli indici di riga e di colonna del suo primo elemento, mentre *dir* viene posto uguale al carattere 'v' se la sequenza è disposta verticalmente, 'o' se orizzontalmente (se la sequenza è presente ripetuta in più posizioni, i valori restituiti possono essere quelli relativi a una qualsiasi delle ripetizioni); se la sequenza non è presente, *riga* e *col* valgono entrambi 0 e *dir* vale 'n'.

- 1. Per codificare la funzione in questione, si sviluppino prima le due seguenti funzioni ausiliarie**
function [pres] = verificaSeqOrizzontaleDaPosizione (matrice, seq, riga, col)
function [pres] = verificaSeqVerticaleDaPosizione (matrice, seq, riga, col)
che ricercano la sequenza nella matrice a partire da una posizione d'inizio precisa, in direzione orizzontale o verticale: *riga* e *col* sono il punto di inizio; il risultato *pres* vale 1 se la sequenza è presente, 0 altrimenti.

2. Successivamente si utilizzano queste due funzioni per codificare le due seguenti
function [riga, col] cercaInizioSeqOrizzontale = (matrice, seq)
function [riga, col] cercaInizioSeqVerticale = (matrice, seq)
che ricercano la sequenza in tutta la matrice con disposizione orizzontale e verticale, restituendo in *riga* e *col* le coordinate del punto d'inizio, se la sequenza viene trovata, o il valore 0 altrimenti.
3. Infine si codifica la funzione *cercaInizioSequenza* facendo uso delle due precedenti.

20'

Blocco 2 / 1 / 1 (20')

CODICE 5

Blocco 2 / 1 / 2 (30')

Una matrice Matlab contiene numeri interi. Si vuole progettare una funzione che ricevendo la matrice e un array di numeri interi che rappresenta una sequenza, cerchi tale sequenza all'interno della matrice. La sequenza può essere disposta, nella matrice, verticalmente dall'alto verso il basso od orizzontalmente, da sinistra verso destra. La funzione deve avere la seguente intestazione:

```
function [riga, col, dir] = cercaInizioSequenza(matrice, seq)
```

se la sequenza è presente nella matrice allora *riga* e *col* indicano gli indici di riga e di colonna del suo primo elemento, mentre *dir* viene posto uguale al carattere 'v' se la sequenza è disposta verticalmente, 'o' se orizzontalmente (se la sequenza è presente ripetuta in più posizioni, i valori restituiti possono essere quelli relativi a una qualsiasi delle ripetizioni); se la sequenza non è presente, *riga* e *col* valgono entrambi 0 e *dir* vale 'n'.

1. Per codificare la funzione in questione, si sviluppino prima le due seguenti funzioni ausiliarie
function [pres] = verificaSeqOrizzontaleDaPosizione (matrice, seq, riga, col)
function [pres] = verificaSeqVerticaleDaPosizione (matrice, seq, riga, col)
che ricercano la sequenza nella matrice a partire da una posizione d'inizio precisa, in direzione orizzontale o verticale: *riga* e *col* sono il punto di inizio; il risultato *pres* vale 1 se la sequenza è presente, 0 altrimenti.
2. Successivamente si utilizzano queste due funzioni per codificare le due seguenti
function [riga, col] cercaInizioSeqOrizzontale = (matrice, seq)
function [riga, col] cercaInizioSeqVerticale = (matrice, seq)
che ricercano la sequenza in tutta la matrice con disposizione orizzontale e verticale, restituendo in *riga* e *col* le coordinate del punto d'inizio, se la sequenza viene trovata, o il valore 0 altrimenti.
3. Infine si codifica la funzione *cercaInizioSequenza* facendo uso delle due precedenti.

Blocco 2 / 1 / 2 (30')

Una matrice Matlab contiene numeri interi. Si vuole progettare una funzione che ricevendo la matrice e un array di numeri interi che rappresenta una sequenza, cerchi tale sequenza all'interno della matrice. La sequenza può essere disposta, nella matrice, verticalmente dall'alto verso il basso od orizzontalmente, da sinistra verso destra. La funzione deve avere la seguente intestazione:

```
function [riga, col, dir] = cercaInizioSequenza(matrice, seq)
```

se la sequenza è presente nella matrice allora *riga* e *col* indicano gli indici di riga e di colonna del suo primo elemento, mentre *dir* viene posto uguale al carattere 'v' se la sequenza è disposta verticalmente, 'o' se orizzontalmente (se la sequenza è presente ripetuta in più posizioni, i valori restituiti possono essere quelli relativi a una qualsiasi delle ripetizioni); se la sequenza non è presente, *riga* e *col* valgono entrambi 0 e *dir* vale 'n'.

1. Per codificare la funzione in questione, si sviluppino prima le due seguenti funzioni ausiliarie

```
function [pres] = verificaSeqOrizzontaleDaPosizione (matrice, seq, riga, col)
```

```
function [pres] = verificaSeqVerticaleDaPosizione (matrice, seq, riga, col)
```

che ricercano la sequenza nella matrice a partire da una posizione d'inizio precisa, in direzione orizzontale o verticale: *riga* e *col* sono il punto di inizio; il risultato *pres* vale 1 se la sequenza è presente, 0 altrimenti.

2. Successivamente si utilizzano queste due funzioni per codificare le due seguenti

```
function [riga, col] cercaInizioSeqOrizzontale = (matrice, seq)
```

```
function [riga, col] cercaInizioSeqVerticale = (matrice, seq)
```

che ricercano la sequenza in tutta la matrice con disposizione orizzontale e verticale, restituendo in *riga* e *col* le coordinate del punto d'inizio, se la sequenza viene trovata, o il valore 0 altrimenti.

3. Infine si codifica la funzione *cercaInizioSequenza* facendo uso delle due precedenti.

Blocco 2 / 1 / 2 (30')

Una matrice Matlab contiene numeri interi. Si vuole progettare una funzione che ricevendo la matrice e un array di numeri interi che rappresenta una sequenza, cerchi tale sequenza all'interno della matrice. La sequenza può essere disposta, nella matrice, verticalmente dall'alto verso il basso od orizzontalmente, da sinistra verso destra. La funzione deve avere la seguente intestazione:

```
function [riga, col, dir] = cercaInizioSequenza(matrice, seq)
```

se la sequenza è presente nella matrice allora *riga* e *col* indicano gli indici di riga e di colonna del suo primo elemento, mentre *dir* viene posto uguale al carattere 'v' se la sequenza è disposta verticalmente, 'o' se orizzontalmente (se la sequenza è presente ripetuta in più posizioni, i valori restituiti possono essere quelli relativi a una qualsiasi delle ripetizioni); se la sequenza non è presente, *riga* e *col* valgono entrambi 0 e *dir* vale 'n'.

1. Per codificare la funzione in questione, si sviluppino prima le due seguenti funzioni ausiliarie

```
function [pres] = verificaSeqOrizzontaleDaPosizione (matrice, seq, riga, col)
```

```
function [pres] = verificaSeqVerticaleDaPosizione (matrice, seq, riga, col)
```

che ricercano la sequenza nella matrice a partire da una posizione d'inizio precisa, in direzione orizzontale o verticale: *riga* e *col* sono il punto di inizio; il risultato *pres* vale 1 se la sequenza è presente, 0 altrimenti.

2. Successivamente si utilizzano queste due funzioni per codificare le due seguenti

```
function [riga, col] cercaInizioSeqOrizzontale = (matrice, seq)
```

```
function [riga, col] cercaInizioSeqVerticale = (matrice, seq)
```

che ricercano la sequenza in tutta la matrice con disposizione orizzontale e verticale, restituendo in *riga* e *col* le coordinate del punto d'inizio, se la sequenza viene trovata, o il valore 0 altrimenti.

3. Infine si codifica la funzione *cercaInizioSequenza* facendo uso delle due precedenti.

Blocco 2 / 1 / 2 (30')

CODICE 6

Blocco 2 / 1 / 2 (20')

```
% Scrivere un programma che chieda all'utente di inserire un
numero positivo (in caso contrario ripete la richiesta)

% 1. verifica se il numero e' perfetto.
% 2. Nel caso non sia perfetto, acquisisce un nuovo numero
% positivo e verifica se i due numeri acquisiti sono amici.
%
% Un numero si dice perfetto quando e' uguale alla somma di tutti
% i suoi divisori escluso se' stesso.
%
% In matematica, sono numeri amicabili o amici quelli per cui la
% somma dei divisori propri di uno e' uguale all'altro e
% viceversa.

% faccio 4 funzioni
% function n=inserisciNumero()
% function res=controllaSePerfetto(n)
% function res=controllaSeAmici(n, m)
% function somma=sommaDivisori(n)
```

Blocco 2 / 1 / 2 (20')

CODICE 6

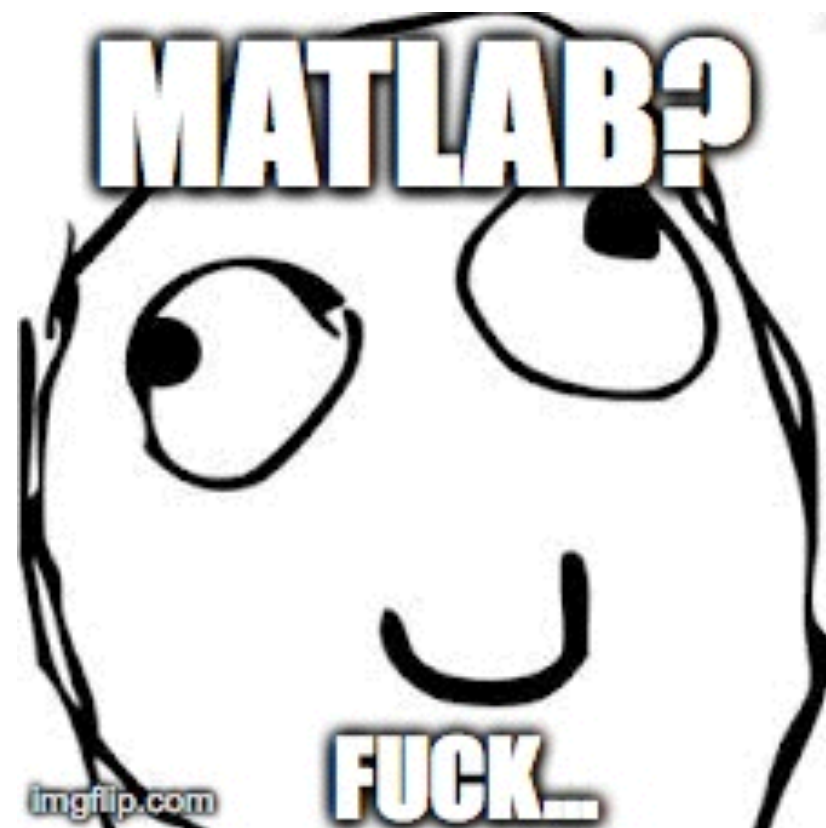
Ottava esercitazione: agenda

- (55') Blocco 1: funzioni MATLAB
- (5') Pausa + domande
- (50') Blocco 2: funzioni MATLAB (livello-esame)
- (10') Pausa + domande
- (10') Question time
- (30') Blocco 3: funzioni ricorsive

Ottava esercitazione: agenda

- (55') Blocco 1: funzioni MATLAB
- (5') Pausa + domande
- (50') Blocco 2: funzioni MATLAB (livello-esame)
- (10') Pausa + domande
- (10') Question time
- (30') Blocco 3: funzioni ricorsive

Question Time



Ottava esercitazione: agenda

- (55') Blocco 1: funzioni MATLAB
- (5') Pausa + domande
- (50') Blocco 2: funzioni MATLAB (livello-esame)
- (10') Pausa + domande
- (10') Question time
- (30') Blocco 3: funzioni ricorsive

Blocco 3 / 1 (15')

```
% Scrivere una funzione che riceve un vettore di interi e :  
% 1) se ha 3 o meno elementi, controlla se il primo di essi  
%    e' dispari:  
%    a) se lo é, lo ritorna  
%    b) se non lo é, lo eleva al quadrato e lo ritorna  
% 2) se ha 4 o più elementi, richiama se stessa accorciando il  
%    vettore levando un elemento dal vettore. Questa sia la  
%    rimozione R-esima:  
%    a) se R é pari, rimuovere il primo valore  
%    b) se R é dispari, rimuovere l'ultimo
```

Blocco 3 / 1 (15')

CODICE 7

Blocco 3 / 2 (15')

```
% Rappresentiamo un polinomio con un vettore contenente i suoi
% coefficienti, dal termine di grado massimo a quello di grado
% minimo. Si noti che un polinomio di grado n corrisponde a un
% vettore di lunghezza n+1

% Esempio:  $3x^4 + 5x^2 + 2x + 7$  (grado 4)
% corrisponde al vettore [3 0 5 2 7] (lunghezza 5).

% Scrivere una funzione ricorsiva di nome derivata che:
% 1) Riceve in ingresso un vettore che rappresenta un polinomio
%     e un valore n
% 2) Restituisca un vettore che rappresenta la derivata n-esima
%     del polinomio

% Per calcolare la derivata prima del polinomio applicare la
% comune regola di derivazione per i polinomi
% Calcolare la derivata n-esima come la derivata prima della
% derivata (n-1)-esima
```

Blocco 3 / 2 (15')

CODICE 8