

ESERCITAZIONI DI INFORMATICA B

FUNZIONI IN MATLAB

Stefano Cereda

stefano.cereda@polimi.it

3/12/2019

Politecnico Milano



Definizione di funzione:

```
1 function [r]=pow(x,y)
2     r=1;
3     for ii=[1:y]
4         r = r*x;
5     end
6 end
```

- lista dei *risultati* fra parentesi quadre
- lista degli *argomenti* fra parentesi tonde
- realizza una singola operazione che può essere usata in altri programmi



Scrivere uno script che, ricevuto in ingresso un vettore di interi m , stampi due vettori p ed n contenenti i numeri di m positivi e negativi (rispettivamente).



Scrivere uno script che, ricevuto in ingresso un vettore di interi m , stampi due vettori p ed n contenenti i numeri di m positivi e negativi (rispettivamente).

Scrivere uno script Matlab che, ricevuto in ingresso un vettore di interi m ed un numero x , stampi due vettori v_1 e v_2 contenenti i numeri di m maggiori e minori di x (rispettivamente.)



Scrivere uno script che, ricevuto in ingresso un vettore di interi m , stampi due vettori p e n contenenti i numeri di m positivi e negativi (rispettivamente).

Scrivere uno script Matlab che, ricevuto in ingresso un vettore di interi m ed un numero x , stampi due vettori v_1 e v_2 contenenti i numeri di m maggiori e minori di x (rispettivamente.)

Scrivere una funzione (split) che data una variabile di ingresso m (scalare, vettore o matrice di numeri) ed un numero x restituisce due vettori $vMin$ e $vMax$. $vMin$ contiene tutti gli elementi di m minori di x , mentre $vMax$ gli elementi maggiori di x .



Scrivere uno script Matlab che, ricevuto un numero a , lo converta in binario utilizzando l'algoritmo delle divisioni ripetute.

Modificare lo script per ottenere una funzione che converta in una base a piacere.



Scrivere una funzione Matlab che, ricevuto in ingresso un array numerico *values* e due interi *min* e *max*, restituisca il numero di elementi di *values* compresi fra *min* e *max*.



Scrivere una funzione Matlab che, ricevuto in ingresso un array numerico *values* e due interi *min* e *max*, restituisca il numero di elementi di *values* compresi fra *min* e *max*.

Scrivere una funzione Matlab che, ricevuto in ingresso un array numerico *values*, ed un intero *bins* restituisca un array che rappresenti l'istogramma dei valori di bins. In particolare, l'intervallo di valori contenuti in *values* andrà diviso in *bins* intervalli ed ogni intervallo avrà un elemento nel nuovo array contenente il numero di elementi di *values* che cadono nell'intervallo.



Scrivere una funzione Matlab che, ricevuto in ingresso un array numerico *values* e due interi *min* e *max*, restituisca il numero di elementi di *values* compresi fra *min* e *max*.

Scrivere una funzione Matlab che, ricevuto in ingresso un array numerico *values*, ed un intero *bins* restituisca un array che rappresenti l'istogramma dei valori di *bins*. In particolare, l'intervallo di valori contenuti in *values* andrà diviso in *bins* intervalli ed ogni intervallo avrà un elemento nel nuovo array contenente il numero di elementi di *values* che cadono nell'intervallo.

Se avessimo utilizzato degli script avremmo condiviso il workspace fra i due pezzi di codice, questo potrebbe portare a cancellare delle variabili. Con le funzioni dobbiamo avere un approccio più pulito in cui ogni funzione lavora solo con i parametri d'ingresso e ritorna tutti i valori di cui potremmo avere bisogno.



Si vuole automatizzare la gestione di una biblioteca di un piccolo paese.

Come primo passo si vuole gestire una matrice di dimensioni $N \times 2$ contenente nella prima colonna un identificativo numerico e nella seconda colonna un numero di tessera.

Immaginate l'identificativo numerico come un codice interno alla biblioteca mentre il numero di tessera come il numero di una tessera esistente appartenente (ad esempio il vostro codice persona del politecnico).

ID	Codice Persona
1	123456789
2	987654321
...	...



Si forniscano le funzioni necessarie all'inserimento di una nuova tessera e alla ricerca dell'identificativo numerico dato un numero di tessera (l'identificativo numerico è un numero crescente che parte da 1).

Funzione *inserisci*:

input: matrice Soci e nuovo numero di tessera

output: la matrice Soci

Funzione *cerca*:

input: matrice Soci e numero di tessera da cercare

output: l'identificativo numerico associato al numero tessera



Un'azienda ha installato un sistema che registra gli ingressi e le uscite dei propri dipendenti tramite l'uso di badge. All'entrata e all'uscita, ogni dipendente deve registrarsi passando il badge nel lettore posto all'ingresso. Siete stati incaricati di scrivere in Matlab parte del software per la gestione dei dati acquisiti.



Il programma Matlab è costituito da una serie di funzioni che lavorano sui seguenti dati:

1. informazioni relative a ciascun dipendente. Queste sono organizzate in una struct con i seguenti campi:
 - ID (un numero intero che identifica univocamente ogni dipendente)
 - nome (stringa)
 - cognome (stringa)
2. registro dei dipendenti. Questo è un array di struct che contiene in ogni suo elemento:
 - i campi di un dipendente (ID, nome, cognome)



- la data che corrisponde a un ingresso e a un'uscita del dipendente (si suppone che avvengano nello stesso giorno). Questa data è rappresentata con un numero intero di 8 cifre. Per esempio, il numero 20171223 corrisponde alla data 23 Dicembre 2017
- oraIngresso e oraUscita, che vengono rappresentati con un numero intero che indica l'ora come il numero di minuti trascorsi dalla mezzanotte precedente (ad esempio, le ore 7:30 corrispondono a 450 minuti, infatti $7 * 60 + 30 = 450$)



1. Per verificare se il dipendente può uscire oppure no, scrivere la funzione *verificaUscita* che prende in ingresso il registro, una data nel formato indicato al punto 2, e un ID e controlla se il dipendente corrispondente ha lavorato nella data passata come parametro per almeno 7 ore e 30 minuti (i.e., 450 minuti). La funzione restituisce vero se questa condizione risulta verificata, falso in caso contrario.
2. Scrivere la funzione *badge* che prende in ingresso un dipendente, la struttura registro, la data odierna (rappresentata come indicato al punto 2), un intero contenente l'ora corrente (rappresentata come indicato al punto 2). La funzione *badge* aggiorna registro nel seguente modo: se nella data odierna il dipendente non compare nella struttura registro (ossia l'utente sta entrando



in azienda), la funzione ne aggiunge le informazioni indicando come data e oraIngresso la data e l'ora passate come parametro. Altrimenti, se il dipendente è già presente nel registro alla data odierna (ossia l'utente sta uscendo dall'azienda), la funzione badge aggiorna il campo oraUscita relativo al dipendente con il valore del parametro ora. La funzione *badge* quindi richiama la funzione *verificaUscita* descritta al punto 1 e, se il dipendente può uscire stampa a video 'Arrivederci!', altrimenti 'Errore uscita!'



3. Si supponga che il registro sia stato precedentemente popolato con i dati relativi ai dipendenti di un'azienda. Si scriva un frammento di codice per definire una variabile *dip1* corrispondente al dipendente “Giacomo Puccini” avente ID = 1858. Si invochi inoltre la funzione badge al fine di inserire nel registro il suo ingresso il 3 Febbraio 2017 alle 8:15.



Un comitato sportivo di sci vuole memorizzare le performance degli sportivi che percorrono la più ripida delle piste del comprensorio. Per ogni atleta, la memorizzazione parte da quota $h_0 = 3000m$ e termina a quota $h_{end} = 1000m$, e viene effettuata in 100 punti diversi del tracciato equidistanti tra loro. I dati vengono salvati in:

- la matrice *atleti*, le cui righe rappresentano i diversi atleti monitorati, e le cui colonne rappresentano i vari tempi collezionati sul tracciato (quindi l'elemento (i, j) della matrice contiene il tempo in cui l'atleta i -esimo è transitato per il j -esimo punto della pista);



- il vettore colonna *velocitaMedia*, con stesso numero di righe della matrice *atleti*, che contiene un valore numerico per ogni atleta, indicativo della sua velocità media lungo tutta la pista.



Scrivere in linguaggio Matlab una funzione *dividiMatrice* che:

1. riceva in input una matrice *atletiTot* (con la stessa struttura di *atleti*), un vettore *velocitaM* (con stessa struttura di vettore *velocitaMedia*) e uno scalare soglia;
2. fornisca in output due matrici *atletiOver* e *atletiUnder* (ognuna con la stessa struttura di *atletiTot*). *atletiOver* include solo le righe di *atletiTot* corrispondenti agli elementi di *velocitaM* con valore maggiore o uguale di soglia. *atletiUnder*, invece, include solo le righe di *atletiTot* corrispondenti agli elementi di *velocitaM* con valore minore di soglia.



Scrivere una funzione per controllare se una matrice è simmetrica.



Si implementi in linguaggio MATLAB una funzione chiamata *partition* che prende in ingresso un vettore V di valori numerici non ordinati, e restituisce in uscita un valore numerico m (valore centrale di V) e un nuovo vettore T , definiti nel modo seguente:



- Se V ha lunghezza dispari, m viene posto pari all'elemento in posizione centrale di V , e il vettore T viene costruito dagli elementi di V , disponendo a sinistra di m tutti gli elementi minori dello stesso m e a destra tutti gli elementi maggiori di m . Ad esempio, dato il vettore $V = [6534312]$, la funzione restituisce il valore centrale $m = 4$ e il vettore $T = [3312465]$.
- Se V ha lunghezza pari, il valore m è definito come la media dei due elementi centrali di V , e gli elementi di V vengono disposti in T in modo tale che tutti quelli minori di m precedono tutti quelli maggiori dello stesso numero. In questo caso, non essendo m parte di V , esso non viene inserito in T . Ad esempio, dato il vettore $V = [561234]$, la funzione restituisce il valore centrale



$m = 1.5$ (cioè la media dei due valori in posizione centrale 1 e 2) e il vettore $T = [156234]$.



Si consideri la seguente struttura dati Temperatura:

```
Temperatura.valore = 20;  
Temperatura.scale = 'celsius'
```

Dove il campo *scale* indica se il campo *valore* è da intendersi espresso in gradi celsius o fahrenheit.

Scrivere una funzione che, ricevuta in ingresso una struct Temperatura restituisca una struct in cui il valore di temperatura è espresso secondo l'altra scala.

Si ricorda che:

$$\begin{aligned} \text{fahrenheit} &= \text{celsius} * (9/5) + 32; \\ \text{celsius} &= (\text{fahrenheit} - 32) * (5/9) \end{aligned}$$

