



Laboratorio n°6 del 07-11-2017

Ing. Dario Cogliati



Es1 – Piloti

- Si studino le strutture dati, che permettono di memorizzare le informazioni di 20 piloti che partecipano ad una gara automobilistica di 50 giri. In particolare, vengono memorizzati il nome del pilota, il nome della sua squadra, il numero del pilota in gara, il numero di giri completati ed il tempo realizzato in ciascun giro completato. Si aggiunga anche il miglior tempo sul giro (cioè il tempo minore) effettuato da ciascun pilota nel corso della gara e se il pilota ha effettivamente completato i 50 giri previsti dalla gara ;
 - Scrivere un frammento di programma MATLAB che, per ciascun pilota, calcoli il miglior tempo sul giro e se il pilota ha completato i 50 giri.
 - Scrivere un frammento di programma MATLAB che stampi a video nome, squadra e numero del pilota che (nel corso della gara) ha compiuto i 50 giri previsti e realizzato in assoluto il miglior tempo sul giro. Se nessun pilota ha completato tutti i giri, il frammento di programma dovrà stampare a video il messaggio “Gara ancora in corso, si prega di attendere”



Es2 – Precipitazioni

- Si consideri l'array definito in Matlab e chiamato *dati*. Tale array contiene le informazioni riguardanti le precipitazioni atmosferiche registrate da una stazione meteorologica. In particolare, ogni elemento dell'array *dati* è una struct che contiene un campo numerico *giorno*, un campo numerico *mese*, un campo numerico *anno*, e un campo numerico *mm* che rappresenta la quantità di pioggia (in mm) caduta nella data specificata dai precedenti campi. Ad esempio, per memorizzare nell'array *dati* che il 3 giugno 2013 sono stati registrati 15 mm di pioggia si avrà:

`dati(i) = struct("giorno", 3, "mese", 6, "anno", 2013, "mm", 15);`



Es2 – Precipitazioni

- Si risponda ai seguenti quesiti:
- A. Scrivere l'intestazione di una funzione Matlab *media* che riceva in ingresso l'array *dati* e un parametro numerico *anno*; la funzione deve restituire in uscita un array di 12 elementi che, sulla base del contenuto dell'array *dati*, fornisca il valore medio dei mm di pioggia in ciascun mese dell'anno solare *anno*.
- B. Scrivere una funzione Matlab di ordine superiore *mostra* che riceva in ingresso l'array *dati*, un parametro funzione *f* e un parametro numerico *anno*. Si assuma che *f* abbia la stessa intestazione della funzione *media* al punto A. La funzione *mostra* deve applicare *f* all'array *dati* per calcolare una statistica mese per mese dell'anno solare specificato dal parametro *anno*; inoltre, la funzione *mostra* deve visualizzare i valori di tale statistica mensile mediante un grafico, con il mese in ascissa e il valore della statistica per ciascun mese in ordinata.
- C. Scrivere un esempio di chiamata della funzione *mostra* implementata al punto precedente, con cui si vuole mostrare il grafico delle precipitazioni medie mensili dell'anno 2012, i cui valori sono contenuti in un array *dati* che deve essere caricato con i valori memorizzati in un file in formato Matlab di nome "*dati*".
- D. Scrivere il corpo della funzione Matlab *media* assumendo che il suo parametro contenga per ogni mese degli anni considerati almeno un dato relativo alle precipitazioni.



Es3 - CheckMatrice

- A. Si scriva in linguaggio MATLAB una funzione `checkMatrice` che riceva come parametri: una matrice, un numero di riga, un numero di colonna e un valore di soglia. La funzione restituisce `true` se tutti i valori degli elementi della matrice adiacenti all'elemento dato (identificato dalla riga e dalla colonna in input) sono strettamente minori del valore di soglia.
- B. Si scriva uno script MATLAB che acquisisca da un file di testo `matrice.txt` i valori di una matrice e da tastiera un numero di riga, un numero di colonna e un valore di soglia, e chiami la funzione `checkMatrice` passandole questi parametri. Si stampi quindi a schermo la scritta “successo!” se il risultato della funzione è vero, “fallimento...” in caso contrario.
- C. Si rappresentino l'ambiente di esecuzione (i.e., il workspace) del chiamante (lo script) e del chiamato (la funzione `checkMatrice`) subito prima del momento in cui la funzione termina. Si considerino come valori di matrice, riga, colonna e soglia quelli mostrati qui sotto. Si indichi il valore che la funzione restituirà al chiamante e, conseguentemente, se lo script stamperà “successo!” o “fallimento...” giustificando la risposta :.



Es4 - Palestra

- Una palestra ha installato un sistema che registra gli ingressi e le uscite dei propri clienti tramite l'uso di badge. All'entrata e all'uscita, ogni cliente deve registrarsi passando il badge nel lettore posto all'ingresso. Siete stati incaricati di scrivere in Matlab parte del software per la gestione degli accessi.

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

1. Informazioni relative a ciascun **cliente**. Queste sono organizzate in una struct con i seguenti campi:

- **matricola** (un numero intero che identifica univocamente ogni cliente),
- **nome** (stringa)
- **cognome** (stringa)

2. L' **archivio** dei clienti. Questo è un array di struct che contiene in ogni suo elemento:

- I campi di un cliente (**matricola**, **nome**, **cognome**).
- La **data** che corrisponde a un ingresso e ad un'uscita del cliente (si suppone che avvengano nello stesso giorno). Questa data è rappresentata con un numero intero di 8 cifre decimali. Per esempio, il numero 20171223 corrisponde alla data 23 Dicembre 2017.
- **eserciziTotali** ed **eserciziSvolti**, ognuno rappresentato con un numero intero che indica rispettivamente il numero di esercizi che il cliente deve svolgere e quelli effettivamente svolti dal cliente.



Es4 - Palestra

A) Per stimare l'intensità dell'allenamento di un cliente si implementi la funzione **valutaAllenamento** che analizza **archivio** e riceve una **matricola** ed una **data**. La funzione calcola **intensitaAllenamento** come la percentuale di esercizi svolti (rispetto a quelli previsti) dal cliente corrispondente a **matricola** nella data specificata. La funzione quindi restituisce *true* se **intensitaAllenamento** è superiore o uguale a 0.47 (allenamento intenso), *false* altrimenti.

B) Scrivere la funzione **tessera** che prende in ingresso i dati di un **cliente**, la struttura **archivio**, la data odierna (rappresentata come indicato al punto 2), un intero **esercizi** contenente il numero di esercizi da svolgere. La funzione aggiorna la variabile **archivio** nel seguente modo: se nella data odierna il cliente non compare nella struttura **archivio** (ossia il cliente sta entrando in palestra), la funzione **tessera** ne aggiunge le informazioni all' **archivio** indicando come **data** ed **eserciziTotali** il valore passati come parametro. Altrimenti, se il cliente è già presente nella struttura **archivio** alla data odierna (ossia il cliente sta uscendo dalla palestra), la funzione **tessera** aggiorna il campo **eserciziSvolti** relativo a quel cliente con il valore del parametro **esercizi** e calcola, richiamando la funzione **valutaAllenamento**, se l'allenamento è stato intenso o meno. Se l'allenamento è intenso per il cliente, stampa a video "Ottimo allenamento!", altrimenti stampa a video "Allenamento poco intenso!".

C) Si supponga che l' **archivio** sia stato precedentemente riempito con i dati dei clienti di una palestra. Si scriva un frammento di codice per definire una variabile **cliente1**, corrispondente al cliente " *Giuseppe Verdi*" avente matricola uguale a 1813. Si invochi quindi la funzione **tessera** al fine di inserire nella variabile **archivio** il suo ingresso il 6 Marzo 2017 con un numero di esercizi da svolgere pari a 40.