

# Esercitazioni di Informatica B

## Struct in MATLAB

---

Stefano Cereda

[stefano.cereda@polimi.it](mailto:stefano.cereda@polimi.it)

27/11/2018

Politecnico di Milano



Data la seguente tabella:

	op1	op2	op3	op4	op5
Paga oraria	5	5.5	6.5	5	6.25
Ore settimanali	40	43	37	50	45
Pezzi prodotti	1000	1100	1000	1200	1100

Scrivere uno script Matlab che organizzi la tabella in un array di dati strutturati e risponda alle seguenti domande:

1. Quanto guadagna ogni operaio?
2. Qual è il salario totale di tutti gli impiegati?
3. Quanti pezzi vengono prodotti?
4. Qual è il costo medio di un pezzo?
5. Quante ore occorrono in media per un pezzo?
6. Qual è l'operaio più efficiente?

Si vogliono rappresentare informazioni relative ai mammiferi ospitati in un parco protetto.

In particolare, per ogni mammifero si rappresentano i seguenti dati:  
NOME TIPO, TIPO ALIMENTAZIONE, PESO MEDIO, insieme di ESEMPLARI

Dove ogni esemplare è caratterizzato da:

CODICE ESEMPLARE, ANNO DI NASCITA, NUMERO CUCCIOLI

1. Si crei una variabile mammifero che contenga le informazioni del seguente mammifero:

NOME TIPO: Elefante

TIPO ALIMENTAZIONE: Erbivoro

PESO MEDIO: 2500

ESEMPLARE 1: (Codice: 3, Anno: 1985, Cuccioli: 3)

ESEMPLARE 2: (Codice: 5, Anno: 1989, Cuccioli: 0)

ESEMPLARE 3: (Codice: 8, Anno: 1982, Cuccioli: 0)

2. Chieda all'utente se vuole inserire ulteriori elefanti
3. Si scriva il codice che costruisce, a partire dalla variabile creata al punto precedente, l'archivio degli esemplari di elefante sterili sapendo che un elefante si dice sterile se è nato prima del 1990 e non ha concepito nessun cucciolo.

Si stampi su schermo l'elenco dei codici degli elefanti sterili.

# Treni in stazione i

Si consideri il sistema di gestione dei treni in una stazione. In ogni stazione ci sono varie banchine e ogni banchina consente la sosta e il transito di un treno. Ad ogni treno in stazione viene assegnato uno dei seguenti stati:

- fuoriStazione: il treno, in attesa di essere assegnato a una banchina, si trova fuori dalla stazione
- inIngresso: il treno è stato assegnato a una banchina ed è in marcia a velocità ridotta per entrare in stazione
- inSosta: il treno è fermo a una banchina
- attesaOut: il treno è fermo a una banchina in attesa di poter partire
- inUscita: il treno sta abbandonando la banchina procedendo a velocità ridotta.

## Treni in stazione ii

Si assuma che siano già stati introdotti i tipi `Stazione` e `Banchina`. In particolare, `Stazione` contiene l'insieme delle banchine presenti in stazione (`banchine`), la coda dei treni in attesa all'ingresso (`codaTreni`) e una variabile Booleana che dice se c'è qualche treno in manovra in stazione (`bloccata`).

`Banchina`, invece, contiene il numero della banchina e i dati del treno in sosta, se la banchina è in stato occupato. Ciascun treno, oltre ad avere il proprio stato e il numero della banchina eventualmente assegnata, ha un campo `minutiAttesaOut` che indica da quanti minuti è in attesa in stato `attesaOut`.

Si supponga che sia stata definita la variabile `stazioneMI` di tipo `Stazione` e che questa variabile sia stata riempita in modo opportuno per rappresentare lo stato corrente della stazione di Milano, si definisca uno script che, se la stazione non è bloccata dalle manovre

di qualche treno e se vi è almeno un treno in coda per entrare in stazione, blocca la stazione, estrae dalla coda il primo treno che vi era stato inserito, gli assegna lo stato inIngresso, e aggiorna la coda in modo tale che poi contenga solo i rimanenti treni in coda. Se per esempio, prima di questa operazione la coda è così costituita:  
stazioneMI.codaTreni(1) = FR2092, stazioneMI.codaTreni(2) = IR097,  
stazioneMI.codaTreni(3) = FA2673 (per semplicità abbiamo riportato solo il nome del treno, ma, ovviamente, codaTreni dovrebbe contenere l'intera struct corrispondente), dopo l'operazione si deve avere che stazioneMI.codaTreni(1)= IR097, stazioneMI.codaTreni(2) = FA2673.