

ESERCITAZIONI DI INFORMATICA B

STRUCT IN MATLAB

Stefano Cereda

stefano.cereda@polimi.it

26/11/2019

Politecnico Milano



Utilizzare gli array logici per risolvere i seguenti esercizi della scorsa esercitazione:

- Confronti — slide 4
- Confronti con matrici — slide 5
- Soglia intorno — slide 10
- Caffè — slide 11
- Matrice dominante — slide 13



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.



Risolvere in MATLAB l'esercizio sulla stazione ferroviaria presentato nelle slide di riepilogo sul C.

Lo script controlla se la stazione non è bloccata dalle manovre di qualche treno e se vi è almeno un treno in coda per entrare in stazione, quindi 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.

