

 POLITECNICO DI MILANO

Dipartimento di
Elettronica e Informazione

Nona esercitazione

Riccardo Cattaneo

Alessandro A. Nacci



POLITECNICO
DI MILANO

17/01/2014

1



Ottava esercitazione: agenda

- (30') **Blocco 0: funzioni base**
- (60') Blocco 1: funzioni di ordine superiore
- (30') Blocco 2: funzioni ricorsive
- (10') Pausa + domande
- (10') Question time
- (30') Blocco 3: funzioni MATLAB (livello-esame)

Blocco 0 / 1 (30')

```
% Realizzare in MATLAB il gioco forza 4
%
% Il gioco prevede due avversari che giocano alternativamente ad inserire in una
% matrice posta verticalmente sul piano di gioco una pedina del proprio colore.
% Vince il giocatore che per primo allinea 4 pedine del proprio colore in orizzontale,
% in verticale od in diagonale.
%
% Si chiede di scrivere 8 funzioni:
% 1) cambiaGiocatore: dato un numero che rappresenta il giocatore 1, restituire il
% numero che rappresenta il giocatore 2 e viceversa
% 2) inserisciPedina: data una matrice parzialmente riempita, la funzione chiede
% ripetutamente al giocatore corrente di inserire una pedina finchè il valore
% che inserisce - la colonna - non é corretto, quindi restituisce la altezza
% del primo buco libero
% 3) richiediColonna: una funzione richiamata da inserisciPedina che richiede
% ripetutamente all utente di inserire un valore di colonna valido
% 4) verificaMatrice: verifica che una matrice contenga una quadrupla in
% orizzontale, verticale e diagonale. Scrivere una funzione per ognuno dei
% tre casi
% 5) forzaQuattro: il "main" del gioco. Fa alternare i giocatori, verifica Loiacono
% stato del gioco, dichiara chi ha vinto
```

Blocco 0 / 1 (30')

CODICE 0

Ottava esercitazione: agenda

- (30') Blocco 0: funzioni base
- (60') Blocco 1: funzioni di ordine superiore
- (30') Blocco 2: funzioni ricorsive
- (10') Pausa + domande
- (10') Question time
- (30') Blocco 3: funzioni MATLAB (livello-esame)

Blocco 1 / 1 (30')

```
% Facendo uso della funzione di ordine superiore accumulatore
% Codificare la funzione tmp(v)
%
% INPUT vettore di numeri v=[v1,..., vn]
%
% OUTPUT Calcola e restituisce come risultato il valore:
%         PRODUTTORIA{1,n} v_i * SOMMATORIA{1,n} v_i
%
% PRODUTTORIA e SOMMATORIA vengono calcolati come termini isolati,
% ottenuti da una opportuna chiamata di una funzione, detta
% accumulatore, che prende una funzione (prodotto o somma) ed i
% limiti tra cui applicare produttoria e sommatoria, e restituisce i
% valori parziali accumulati
```

Blocco 1 / 1 (30')

CODICE 1

accumulatore.m

prod2.m

sum2.m

tmp.m

tmp2.m

Blocco 1 / 2 (30')

```
% Scrivere una funzione O che prenda in ingresso un vettore di interi ed una
% funzione C, nello specifico una funzione di confronto tra due numeri.
% La funzione C puo' confrontare i numeri come preferisce, fintanto che il
% valore restituito (0 o 1) rappresenti la relazione di maggiore e minore
% generalizzato, ossia la funzione C sia un ordine parziale.
% O ordina il suo vettore di dati in ingresso nel senso della funzione C.
```

Blocco 1 / 2 (30')

CODICE 2

Ottava esercitazione: agenda

- (30') Blocco 0: funzioni base
- (60') Blocco 1: funzioni di ordine superiore
- (30') Blocco 2: funzioni ricorsive
- (10') Pausa + domande
- (10') Question time
- (30') Blocco 3: funzioni MATLAB (livello-esame)

Blocco 2 / 1 (30')

```
%% FIBONACCI SEQUENCE
% fib(0) = 1
% fib(1) = 1
% fib(n) = fib(n-1) + fib(n-2)
% eseguite tutti gli step intermedi di fib(5)
```

Blocco 2 / 1 (30')

CODICE 3

Blocco 2 / 1 (30')

```
>> fib(5)
sto eseguendo con n=5, per cui invoco fib(4)+fib(3)sto eseguendo con n=4, per cui invoco fib(3)+fib(2)sto eseguendo
con n=3, per cui invoco fib(2)+fib(1)sto eseguendo con n=2, per cui invoco fib(1)+fib(0)sto eseguendo con n=1
n = 1
sto eseguendo con n=0
n = 0
n = 1
sto eseguendo con n=1
n = 1
n = 2
sto eseguendo con n=2, per cui invoco fib(1)+fib(0)sto eseguendo con n=1
n = 1
sto eseguendo con n=0
n = 0
n = 1
n = 3
sto eseguendo con n=3, per cui invoco fib(2)+fib(1)sto eseguendo con n=2, per cui invoco fib(1)+fib(0)sto eseguendo
con n=1
n = 1
sto eseguendo con n=0
n = 0
n = 1
sto eseguendo con n=1
n = 1
n = 2
n = 5
ans = 5
>>
```

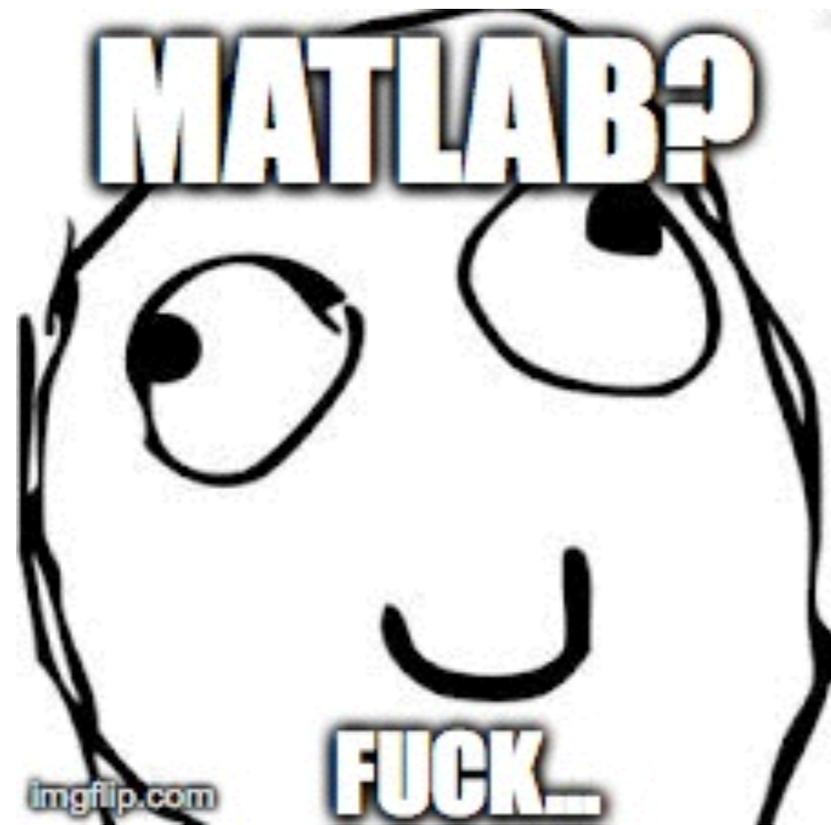
Ottava esercitazione: agenda

- (30') Blocco 0: funzioni base
- (60') Blocco 1: funzioni di ordine superiore
- (30') Blocco 2: funzioni ricorsive
- (10') Pausa
- (10') Question time
- (30') Blocco 3: funzioni MATLAB (livello-esame)

Ottava esercitazione: agenda

- (30') Blocco 0: funzioni base
- (60') Blocco 1: funzioni di ordine superiore
- (30') Blocco 2: funzioni ricorsive
- (10') Pausa + domande
- (10') Question time
- (30') Blocco 3: funzioni MATLAB (livello-esame)

Question Time



Ottava esercitazione: agenda

- (30') Blocco 0: funzioni base
- (60') Blocco 1: funzioni di ordine superiore
- (30') Blocco 2: funzioni ricorsive
- (10') Pausa + domande
- (10') Question time
- (30') Blocco 3: funzioni MATLAB (livello-esame)

Blocco 3 / 1 (15')

Si conviene di rappresentare un'immagine mediante una matrice rettangolare di numeri, corrispondenti al colore dei suoi pixel (punti luminosi che compongono la figura). Si vuole progettare una funzione MATLAB/Octave di nome `combinaImmagini` che, ricevendo come parametri due matrici $f1$ ed $f2$ rappresentanti due immagini e due valori numerici C ed S , con $C < S$, rappresentanti due diversi colori, produce come risultato una terza figura $f3$, ottenuta da $f1$ ed $f2$ secondo la seguente regola.

Nelle posizioni (r, c) in cui $f1(r, c) < C$ ed $f2(r, c) > C$ si ha $f3(r, c) = f2(r, c) - f1(r, c)$ nelle posizioni (r, c) in cui $f1(r, c) > S$ ed $f2(r, c) < S$ si ha $f3(r, c) = f1(r, c) - f2(r, c)$ nelle posizioni rimanenti si ha $f3(r, c) = f1(r, c)$.

- . a) Codificare la funzione `combinaImmagini`.
- . b) Scrivere uno script che acquisisce le due matrici di partenza rispettivamente dai file `file1.mat` e `file2.mat` (si supponga che le due variabili contenenti le matrici al momento del salvataggio si chiamino `matr1` e `matr2`), richiama la funzione `combinaImmagini` e salva sul file `file3.mat` la matrice risultante.

Blocco 3 / 1 (30')

CODICE 4