



# **Informatica B**

## **2016-2016**

### **Esercitazione IV**

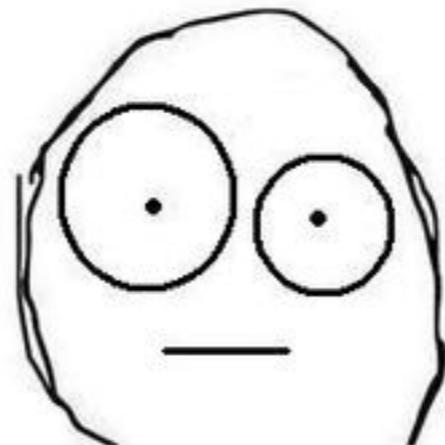
*Variabili funzione & Funzioni di Ordine Superiore*

Alessandro A. Nacci

[alessandro.nacci@polimi.it](mailto:alessandro.nacci@polimi.it) - [www.alessandronacci.it](http://www.alessandronacci.it)

Facciamo un ripasso di quanto visto ieri a lezione perché l'argomento necessita di freschezza di memoria....

La vostra, non quella del PC ;)



**Wait. WHAT?**

## Variabili funzioni

- ❑ Matlab permette di assegnare a variabili valori di tipo "funzione"
- ❑ Un valore di tipo funzione può essere assegnato a una variabile (quindi passarlo come parametro), detta **handle**
- ❑ L'handle può essere applicato a opportuni argomenti per ottenere una invocazione della funzione

## Assegnamento di un valore di tipo funzione

- Handle di una funzione esistente

```
f = @nome_funzione
```

- ▶ Esempio

```
>> seno=@sin
seno = @sin
>> seno(pi/2)
ans = 1
```

- Handle di una funzione definita ex-novo

```
f = @(x,y...) <expr>
```

- ▶  $x, y, \dots$  sono i parametri della funzione
- ▶  $\langle \text{expr} \rangle$  è un'espressione che calcola il valore della funzione
- ▶ Esempio

```
>> sq=@(x) x^2
sq = @(x) x^2
>> sq(8)
ans = 64
```

## Funzioni di ordine superiore

- ❑ Se un parametro di una funzione  $f$  è un handle (cioè contiene un valore di tipo funzione) allora  $f$  è una **funzione di ordine superiore**
- ❑ L'handle passato come parametro consente ad  $f$  di invocare la funzione passata
- ❑ Esempio: funzione `map` che applica una funzione  $f$  a tutti gli elementi contenuti nel parametro `vin` e ritorna i risultati in `vout`

```
function [vout]=map(f, vin)
    for i=1:length(vin)
        vout(i)=f(vin(i));
    end;
```

handle

```
>> A=[1,2,3,4,5,6];
>> map(sq,A)
ans = 1 4 9 16 25 36
```

Invoca la funzione passata come argomento

## Esempio: funzione accumulatore

- ❑ Funzione accumulatore: `[x] =acc(f, a, u)`
  - ▶ applica cumulativamente l'operazione binaria `f` (con elemento neutro `u`) a tutti gli elementi di `a`:

`f(...f(f(f(u,a(1)),a(2)),a(3))...,a(length(a)))`

```
function [x]=acc(f, a, u)
    x=u;
    for i=1:length(a)
        x=f(x, a(i));
    end
```

- ❑ Funzione sommatoria: `function [s]=sommatoria(a)`
  - ▶ calcola la sommatoria degli elementi di `a`

```
function [s]=sommatoria(a)
    som=@(x,y)x+y;
    s=acc(som,a,0);
```

Nel gioco della roulette si ha una ruota con 37 numeri da 0 a 36.

- Si scriva una funzione che simuli la ruota e generi il numero vincente accettando come parametro un generatore di numeri casuali compresi tra 0 e 36.
- Si scriva una funzione che provi a eseguire N giri e salvi i risultati delle frequenze di ogni numero in un array di 37 elementi (il primo elemento per lo 0, il secondo per l'1 e così via). La funzione riceverà in ingresso anche il generatore di numeri casuali.
- Si supponga di ripetere il calcolo precedente con una roulette truccata che ha probabilità doppia di far uscire uno 0 rispetto a qualsiasi altro numero.
- Rappresentare con un grafico l'array dei 37 elementi e calcolare la probabilità del risultato





# Esercizio I

Nel gioco della roulette si ha una ruota con 37 numeri da 0 a 36.

- Si scriva una funzione che simuli la ruota e generi il numero vincente accettando come parametro un generatore di numeri casuali compresi tra 0 e 36.

```
function [ num ] = es4_estrazione( generatore )
```

```
    num = generatore();
```

```
end
```

```
function [ num ] = generatore()
```

```
    num = randi(37) - 1;
```

```
end
```

$X = \text{randi}(imax)$  returns a pseudorandom scalar integer between 1 and  $imax$ .



# Esercizio I

- Si scriva una funzione che provi a eseguire N giri e salvi i risultati delle frequenze di ogni numero in un array di 37 elementi (il primo elemento per lo 0, il secondo per l'1 e così via). La funzione riceverà in ingresso anche il generatore di numeri casuali.

```
function [ result ] = es4_tutteEstrazioni( N, generatore )

    result = zeros(1,37);

    for i = 1:N
        num = generatore();
        result(num+1) = result(num+1) + 1;
    end

end
```



# Esercizio I

- Si supponga di ripetere il calcolo precedente con una roulette truccata che ha probabilità doppia di far uscire uno 0 rispetto a qualsiasi altro numero.

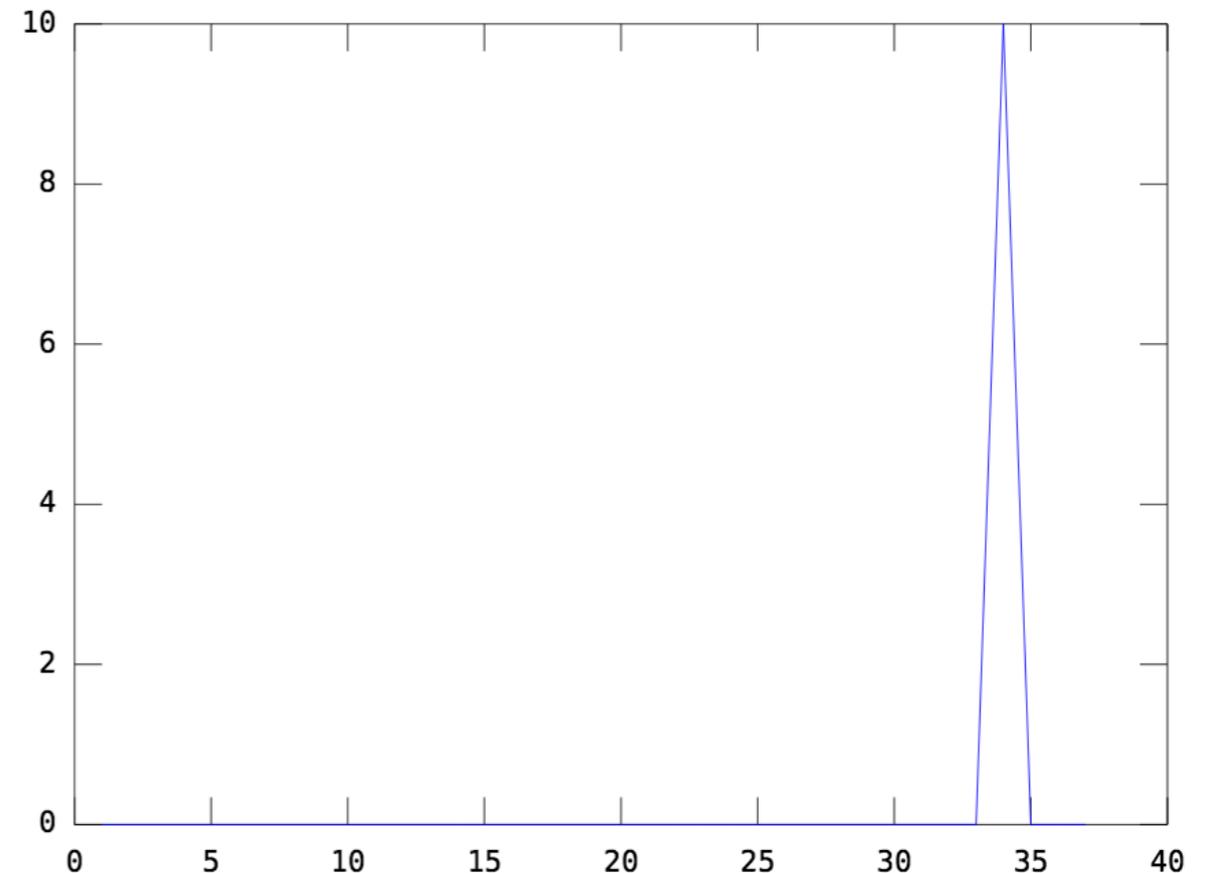
```
function [ num ] = generatore()  
  
    n = rand();  
  
    if (n>0.3)  
        num = 0;  
    else  
        num = generatoreFair();  
    end
```



# Esercizio I

- Rappresentare con un grafico l'array dei 37 elementi e calcolare la probabilità del risultato

```
function [ result ] = es4_calcoloProbabilita( N, generatore )  
  
    result = zeros(1,37);  
  
    for i = 1:N  
        num = generatore();  
        result(num+1) = result(num+1) + 1;  
    end  
  
    x_axes = [1:37];  
  
    plot(x_axes, result);  
  
    result = result / N;  
  
end
```



- Si consideri la seguente funzione di ordine superiore:

```
function r = fun(v,f)
    r = v(1);
    for i = 2:length(v)
        r = f(r,v(i));
    end
```

- Qual è il valore ritornato dalla chiamata `fun([8 9 10 9 6 0],@max)`? Giustificare la risposta. **max** è una funzione offerta dalla libreria di Matlab. In particolare, se  $X$  e  $Y$  sono due scalari,  $\max(X,Y)$  ritorna il maggiore tra i due valori.

**La fun([8 9 10 9 6 0],@max) restituisce il valore 10.**

- Dire che cosa fa la funzione `fun` quando l'argomento  $v$  è un vettore numerico (di lunghezza pari almeno ad 1) ed  $f$  è `@max`

**La funzione restituisce l'elemento di maggior valore di  $v$ .**

- Implementare una versione ricorsiva della funzione `fun` che non richieda di utilizzare cicli (`for` o `while`). L'intestazione della funzione deve rimanere invariata.



## Esercizio 2

- Implementare una versione ricorsiva della funzione fun che non richieda di utilizzare cicli (for o while). L'intestazione della funzione deve rimanere invariata.

```
function r = fun(v,f)
    r = v(1);
    for i = 2:length(v)
        r = f(r,v(i));
    end
```

```
function r = fun(v,f)
    n = length(v);
    if n == 1
        r = v;
    else
        r = f(v(1), fun(v(2:end), f));
    end
```

**Tutte il materiale sarà  
disponibile sul mio sito  
internet!**

[www.alessandronacci.it](http://www.alessandronacci.it)

**See You Next Time!**

