



Politecnico di Milano

Facoltà di Ingegneria Civile, Ambientale e Territoriale

Informatica, Prof. G. Boracchi  
Allievi Ingegneria Civile e Mitigazione del Rischio

Esame del 22 Gennaio 2019

Cognome e nome	
Matricola	
Firma	

Domanda	1	2	3	4	TOT
Punteggio max	10	10	6	6	32
Punteggio					

La **durata** della **prova** è di **due ore e mezza**. Non è consentito consultare libri o appunti, non è consentito l'uso di calcolatrici.

Scrivere solo sui fogli distribuiti utilizzando il retro delle pagine in caso di necessità e cancellando le parti di brutta con un tratto di penna. Non separare questi fogli.

Per tutti gli esercizi non è sufficiente fornire il risultato, ma è **necessario mostrare il procedimento seguito**. **Non dilungarsi però in spiegazioni prolisse**.

Gli allievi sono invitati a privilegiare **chiarezza, proprietà di linguaggio e sinteticità** nelle risposte agli esercizi, con l'obiettivo di **dimostrare la loro conoscenza degli argomenti**.

### Domanda 1. Matlab

Si definisca una funzione `calcolaCorrispondenze` che riceve in input due matrici `m1` e `m2` di  $N \times N$  interi. La funzione restituisce tutti gli elementi di `m2` che sono pari alla somma di due qualsiasi elementi di `m1`. Se un dato elemento di `m2` è somma di più coppie di elementi di `m1`, viene contato una volta sola.

1. Si definisca l'intestazione della funzione `calcolaCorrispondenze`
2. Si implementi la funzione `calcolaCorrispondenze`
3. Si scriva un breve script che richiede all'utente di inserire due matrici, invoca la funzione `calcolaCorrispondenze` e quindi stampa a schermo tutti gli elementi

**Esempio:**

```
>> inserire m1 [5 6; 44 1]
```

```
>> inserire m2 [2 3; 4 1]
```

```
i valori corrispondenti sono      5      6
```

**Hint:** utilizzare una funzione ausiliaria che esegue il controllo su un solo elemento per facilitare lo sviluppo e migliorare la comprensione del codice.

## SOLUZIONE

```
m1 = input('inserire m1 ');
m2 = input('inserire m2 ');

disp(m1)
disp(m2)

vet = calcolaCorrispondenze(m1, m2);

disp('i valori corrispondenti sono');

disp(vet)

%%%%%
function vet = calcolaCorrispondenze(m1, m2)

vet = [];
for ii = 1 : size(m1, 1)
    for jj = 1 : size(m1, 2)
        if ~any(vet == m1(ii, jj)) && esisteSomma(m2, m1(ii, jj))
            vet = [vet, m1(ii, jj)];
        end
    end
end

%%%%%
function res = esisteSomma(m, n)

res = false;
ii = 1;
while(res == false && ii <= size(m,1))
    jj = 1;
    while(res == false && jj <= size(m,2))
        kk = 1;
        while(res == false && kk <= size(m,1))
            hh = 1;
            while(res == false && hh <= size(m,2))
                if(m(ii,jj) + m(kk,hh)== n)&& (ii ~= kk || jj ~= hh)
                    res = true;
                end
                hh = hh + 1;
            end
            kk = kk + 1;
        end
        jj = jj + 1;
    end
    ii = ii + 1;
end
```

## Domanda 2. Matlab: Programmazione funzioni con sottovettori

Scrivere una funzione `prodottoPuntuale` che riceve in ingresso due vettori **A** e **B** e ne calcola il prodotto puntuale. Rispetto all'operatore `A.*B`, la funzione `prodottoPuntuale` esegue il prodotto anche quando i vettori non hanno lo stesso numero di elementi. In tal caso, viene restituito un vettore contenente il prodotto tra il vettore più corto e il vettore più lungo troncato alla lunghezza del vettore più corto.

Si scriva una seconda funzione `prodottoPuntuale2` che restituisce i valori del prodotto puntuale (calcolato come sopra) solo se maggiori di 10 e se pari.

Si scriva infine uno script per richiedere all'utente i due vettori A e B, si invocano su questi:

- `prodottoPuntuale`, salvando il risultato in un vettore **v1**
- `prodottoPuntuale2`, salvando il risultato in un vettore **v2**.

Si visualizzi in un grafico tutti gli elementi di **v1** con una croce rossa e gli elementi di **v2** con una croce rossa ed un cerchio blu

### Esempio

A =

1      2      3      4      5

B =

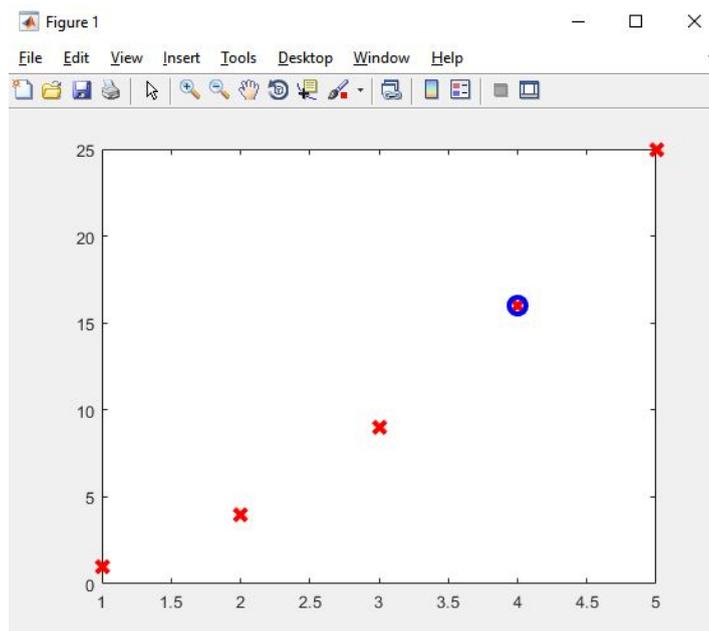
1      2      3      4      5      6      7

v1 =

1      4      9      16      25

v2 =

16



SOLUZIONE

```
clear
```

```
clc
```

```
close all
```

```
A = input('inserire A: ');
```

```
B = input('inserire B: ');
```

```
A
```

```
B
```

```
v1 = prodottoPuntuale(A,B)
```

```
v2 = prodottoPuntuale2(A,B)
```

```
figure(1),
```

```
plot([1 :length(v1)], v1, 'rx', 'MarkerSize', 10, 'LineWidth', 3);
```

```
hold on;
```

```
for ii = 1 : length(v1)
```

```
    if(any(v1(ii) == v2))
```

```
        plot(ii, v1(ii), 'bo', 'MarkerSize', 10, 'LineWidth', 3);
```

```
    end
```

```
end
```

```
%%%
```

```
function C = prodottoPuntuale(A,B)
```

```
maxLen = min(length(A), length(B));
```

```
C = A(1 : maxLen) .* B(1 : maxLen);
```

```
%%%
```

```
function C = prodottoPuntuale2(A,B)
```

```
C = prodottoPuntuale(A,B);
```

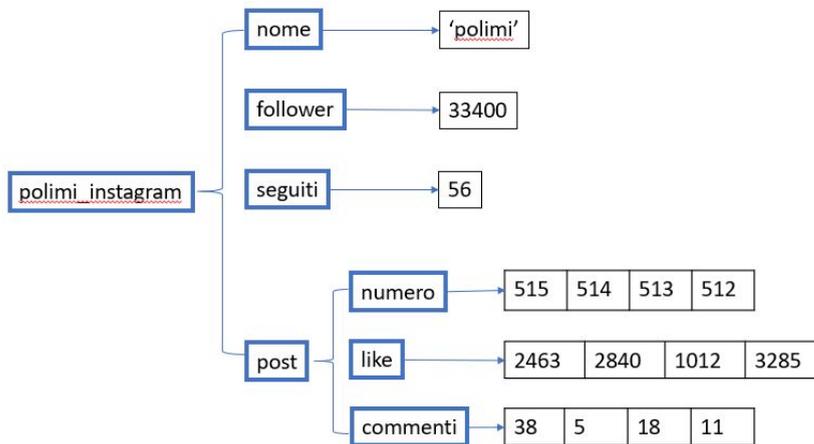
```
C = C(C > 10 & mod(C, 2) == 0);
```

### Domanda 3. Matlab:

Si consideri il seguente codice

```
disp(['Nome account instagram: ',polimi_instagram.nome]);  
disp(['Numero followers: ',num2str(polimi_instagram.follower)]);  
disp(['Numero di account seguiti: ',num2str(polimi_instagram.seguiti)]);  
vect=[];  
for ind=1:length(polimi_instagram.post(:))  
    if (polimi_instagram.post(ind).commenti>5 &&...  
        polimi_instagram.post(ind).like>1000)  
        vect=[vect polimi_instagram.post(ind).commenti];  
    end  
end  
figure  
bar(vect);
```

E si assuma che le variabili sono popolate come nell'illustrazione seguente



1. Si scriva l'output in Command window prodotto dal codice precedente
2. Scrivere il contenuto del vettore `vect`
3. Tracciare il grafico prodotto dall'ultima riga del codice

#### Domanda 4.

Un'abitazione è dotata di un impianto di riscaldamento controllato da un termostato programmabile. È possibile impostare l'accensione dell'impianto nelle fasce orarie notturna e diurna, oppure accendere manualmente la caldaia con un interruttore che ha due stati: acceso (=1) e spento (=0).

1. Scrivere una funzione Matlab che riceve in ingresso un'ora (un numero compreso tra 1 e 24) e lo stato dell'interruttore e restituisce 1 se l'ora è compresa tra le 7 e le 20, oppure se l'interruttore è acceso. In alternativa restituisce 0.
2. Si traduca l'espressione sopra in predicati booleani e si scriva la tabella di verità della condizione implementata nella funzione
3. Si assuma che l'orario sia codificato in minuti a partire da 00:00. Si scriva quanti bit occorrono per rappresentare un orario in
  - a. Codifica binaria
  - b. Codifica modulo e segno
  - c. CP2

e si codifichi quindi l'orario 8:40 e 7:20 in CP2 e se ne calcoli la differenza

#### SOLUZIONE

```
1. function out=termostato(H, flag)
    if (H>=7 && H<=20) || flag==1
        out=1;
    else
        out=0;
    End
```

2.

H>=7	H<=20	flag==1	H>=7 && H<=20	(H>=7 && H<=20)    flag==1
0	0	0	0	0
0	0	1	0	1
0	1	0	0	0
0	1	1	0	1
1	0	0	0	0
1	0	1	0	1
1	1	0	1	1
1	1	1	1	1

3. Un orario espresso in minuti è compreso tra 0 e 60\*24=1440.

Codifica binaria: nbit =11

Codifica modulo e segno: nbit=12

CP2: nbit=12

7:20 = 440 minuti

In cp2 con 12 bit: 000110111000

8:40 = 520 minuti

In cp2 con 12 bit: 001000001000

La differenza 8:40-7:20 è: (1)00001010000