



**Politecnico di Milano**

**Facoltà di Ingegneria Civile, Ambientale e Territoriale**

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

**Esame del 20 Febbraio 2019**

<i>Cognome e nome</i>	
<i>Matricola</i>	
<i>Firma</i>	

<b>Domanda</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>TOT</b>
<b>Punteggio max</b>	<b>12</b>	<b>12</b>	<b>4</b>	<b>4</b>	<b>32</b>
<b>Punteggio</b>					

La **durata** della **prova** è di **2 ore**. 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.

Si progetti e codifichi la funzione **function res = numSommaDiff(a)**, che riceve in ingresso un array **a** di interi.

La funzione restituisce **res = 1** se esiste almeno un elemento in **a** pari alla somma degli elementi che lo seguono diminuita della somma degli elementi che lo precedono. Altrimenti restituisce 0.

**Ad esempio**, nel vettore **1 2 1 20 -6 16 14**, il 20 soddisfa la proprietà descritta, perché la somma degli elementi che seguono è **-6 + 16 + 14 = 24**. La somma degli elementi che precedono è **4**, quindi **20 = 24 - 4** soddisfa la condizione. Quindi la funzione restituisce 1 se invocata su questo vettore.

Se invece si considera il vettore **1 2 1 20 -14 20 14** nessun numero soddisfa la proprietà descritta. Quindi la funzione restituisce 0.

## SOLUZIONE

```
function res = numsommaDiff(a)

res = 0;
for ii = 1 : length(a)
    if ii > 1
        sommaPrecedenti = sum(a(1 : ii - 1));
    else
        sommaPrecedenti = 0;
    end

    if ii < length(a)
        sommaSeguenti = sum(a(ii + 1 : end));
    else
        sommaSeguenti = 0;
    end

    if a(ii) == (sommaSeguenti - sommaPrecedenti)
        res = 1;
        break;
    end
end
```

## Domanda 2.

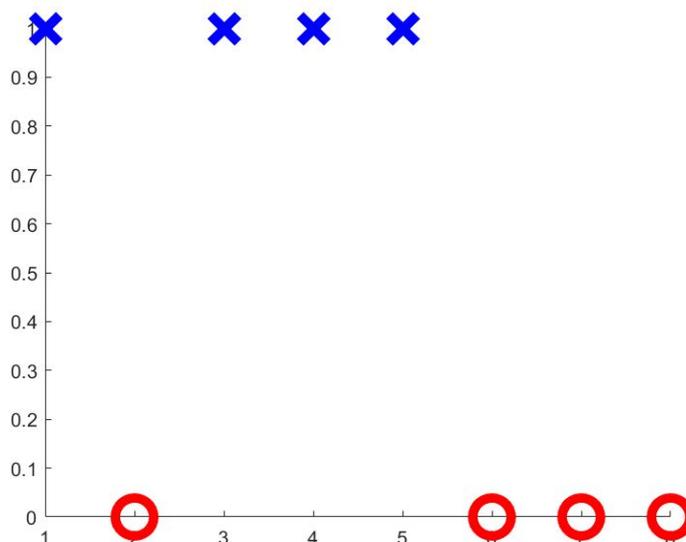
La porta di un'abitazione è dotata di un sistema di apertura automatizzato tramite un codice di 4 cifre.

- Scrivere in linguaggio Matlab una funzione chiamata **'apriporta'**, che riceve lo stato della porta (0 = chiusa, 1 = aperta) e una struttura dati, chiamata **porta\_struct** così composta:
  - `porta_struct.codice=1234`, è il codice segreto che permette di aprire la porta
  - `porta_struct.registro=[0 1 1 1 1]`; qui vengono memorizzati i casi in cui la porta è stata aperta e quando è scattato l'allarme. (1 = porta aperta, 0 = allarme scattato)

La funzione **'apriporta'** richiede all'utente di inserire il codice di apertura, e restituisce una variabile **'apri'** uguale a 1 se lo stato della porta risulta chiuso e se il codice inserito dall'utente è uguale a quello presente nella struttura dati. In questo caso, il registro deve essere aggiornato inserendo alla fine del vettore '1'. Si faccia in modo che la modifica alla struttura dati venga propagata nel workspace principale.

Se invece il codice inserito risulta essere errato, la funzione continua a richiedere di inserire il codice. Se il codice viene inserito sbagliato per più di tre volte scatta l'allarme (allarme=1). In questo caso, la funzione restituisce `apri=0` e la struttura dati aggiornata, dove viene aggiunto un valore 0. Si faccia in modo che la modifica alla struttura dati venga propagata nel workspace principale.

- Scrivere uno script Matlab che invoca:
  - Carica dal file `porta.mat` la struttura **porta\_struct** opportunamente popolata
  - Invoca la funzione **'apriporta'** modificando la struttura **porta\_struct** caricata
  - Visualizza il registro degli accessi come mostrato nella figura sottostante; dove le croci blu rappresentano le volte in cui la porta è stata aperta, mentre i cerchi rossi rappresentano le volte in cui l'allarme è scattato.





## Soluzione

```
function [apri,porta_struct]=apriporta(stato_porta,porta_struct)
cont_errore=0;
apri=0;
allarme=0;
while allarme==0 && stato_porta==0
    codice=input('Inserire codice');
    if codice==porta_struct.codice
        stato_porta=1;
        apri=1;
        porta_struct.registro=[porta_struct.registro, 1];
    else
        cont_errore=cont_errore+1;
        if cont_errore>3
            allarme=1;
            porta_struct.registro=[porta_struct.registro, 0];
        end
    end
end
End

clear
close all
clc
load porta_struct
[apri,porta_struct]=apriporta(0,porta_struct);
figure(1)
hold on
for x=1:length(porta_struct.registro)
    if porta_struct.registro(x)==1
        plot(x,1,'bx','MarkerSize',20,'LineWidth',5);
    else
        plot(x,0,'ro','MarkerSize',20,'LineWidth',5);
    end
end
hold off
save porta_struct
```

### Domanda 3

Si consideri la seguente funzione

```
function d = pippo(s, M, a)  
  
if(s(1) == M(a))  
    if length(s) == 1  
        d = 1;  
        return  
    end  
    d = pippo(s(2 : end), M, a + 1);  
else  
    d = 0;  
end
```

Si riporti quanto visualizzato dalle seguenti invocazioni

```
>> M = 'ciaoMamma';  
>> s = 'amm';  
>> disp(pippo(s, M, 3))
```

```
>> s = 'amm'  
>> disp(pippo(s, M, 6))
```

```
>> s = 'oMama'  
>> disp(pippo(s, M, 4))
```

Si descriva brevemente cosa fa' la funzione pippo (max 3 righe)

---

---

---

**SOL**

La funzione controlla se la stringa  $s$  è contenuta nella stringa  $M$  a partire dalla posizione  $a$

**Domanda 4.****CP2:**

- Convertire in cp2 i seguenti numeri :

 $78_{10}$  $79_{10}$ 

- Specificare il numero minimo di bit necessario per rappresentare correttamente i due numeri.
- Eseguire la somma dei due numeri, indicando i valori dei bit di carry e overflow.

### Tabella di verità:

La **funzione1**, presentata di seguito, riceve due input numerici A e B e restituisce la variabile out. Estrapolare il predicato presente all'interno di **funzione1** per cui viene assegnato out=1 e prepararne la tabella di verità.

```
function out=funzione1(A,B)
```

```
out=0;
```

```
while A>0
```

```
    if B>=5
```

```
        if B*A==50
```

```
            out=1;
```

```
        end
```

```
    end
```

```
end
```

### Soluzione

78=01001110

79=01001111

nbit=8

78+79=[1](0)10011101

Predicato

$B \geq 5 \ \&\& \ B * A == 50 \ \&\& \ A > 0$

A>0	B>=5	B*A==50	B>=5 && B*A==50	B>=5 && B*A==50 && A>0
1	0	0	0	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

La seconda parte della tabella dove A>0 è falsa non la riporto visto che non viene eseguita dal codice.