



**Politecnico di Milano**

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

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

**Esame del 6 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. Programmazione

Scrivere una funzione che riceve due stringhe: *parola* e *elimina*. La funzione cerca in *parola* tutti i caratteri che compongono la stringa *elimina*, e li rimuove solamente se li trova tutti e nell'ordine in cui compaiono in *elimina*, anche se non consecutivi. L'operazione viene ripetuta finché in *parola* è contenuta un'intera istanza della stringa *elimina*. La funzione, oltre a modificare l'array *parola*, restituisce il numero di volte che ha eliminato i caratteri dell'intera stringa *elimina* da *parola*. Nell'eliminare caratteri non devono essere lasciati buchi, ma *parola* dev'essere ricompattata.

Esempio

parola: amaarrreeemmmarrreeaaaarrmae

elimina: mare

La funzione trova tutte le lettere di "mare" in "aMAaRrrEeeemmmarrreeaaaarrmae" che diventa "aarreeemmmarrreeaaaarrmae"

La funzione trova tutte le lettere di "mare" in "aarreeeMmmmARrrEeaaaarrmae" che diventa "aarreeemmrreaaaarrmae"

La funzione trova tutte le lettere di "mare" in "aarreeemmMrreAaaaRrrmaE" che diventa "aarreeemmrreaaaarrma"

Poi non trova più tutte le lettere e restituisce 3

**Hint:** adottare un'opportuna sottofunzione per facilitare lo sviluppo e la comprensione del codice.

## Domanda 2. Programmazione

*Giochiamo a Tris!*

Scrivere in linguaggio Matlab una funzione *controllaTris* che riceve in ingresso una matrice  $M$  di dimensioni  $3 \times 3$  che rappresenta lo schema del gioco del tris e restituisce

- 1 se Giocatore 1 ha fatto tris,
- 2 se Giocatore 2 ha fatto tris,
- 0 se nessuno ha fatto tris

Per fare tris si intende riempire tre caselle adiacenti in orizzontale, in verticale o in diagonale.

Lo schema del tris è così rappresentato nella matrice  $M$ . La cella  $M(i, j)$  contiene:

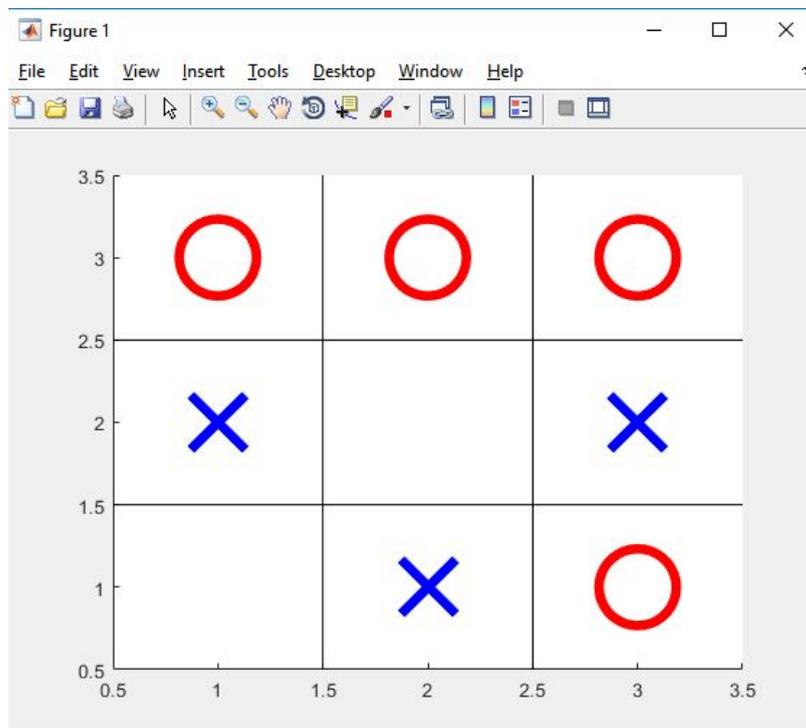
- '0' se la casella nella riga  $i$ , colonna  $j$  è vuota
- '1' se la casella nella riga  $i$ , colonna  $j$  è occupata dal Giocatore 1
- '2' se la casella nella riga  $i$ , colonna  $j$  è occupata dal Giocatore 2

HINT:

Scrivere delle sotto-funzioni che controllano righe, colonne e diagonali.

Richiesta:

- Si scriva la funzione ***controllaTris*** che svolge le operazioni sopra descritte
- Si scriva una funzione ***visualizzaTris*** che prende in ingresso la matrice  $M$  e visualizza in una figura, mediante il comando `plot`, la matrice del tetris come sotto illustrato



## SOLUZIONE

```
%% Controlla tris
mat=[1 0 1
      1 1 2
      1 0 0];
%%
function risultato=controlla_tris(mat)
jj=1;
out_c=0;
out_r=0;
while jj<=size(mat,1) && out_r~=1 && out_c~=1
    out_r=controlla_riga(mat(jj,:)); %controllo riga
    out_c=controlla_riga(mat(:,jj)); %controlla colonna
    jj=jj+1;
end
out_d=controlla_diagonale(mat);
if out_r==1 || out_c==1 || out_d==1
    risultato=1;
elseif out_r==2 || out_c==2 || out_d==2
    risultato=2;
else
    risultato=0;
end

%%
function out=controlla_diagonale(mat)
antidiag=[];
diag=[];
vect=size(mat,1):-1:1;
for jj=1:length(mat)
    for ii=1:length(mat)
        if jj==ii
            diag=[diag mat(jj,ii)];
            antidiag=[antidiag mat(jj,vect(ii))];
        end
    end
end
end
if all(antidiag==1) || all(diag==1)
    out=1;
elseif all(antidiag==2) || all(diag==2)
    out=2;
else
    out=0;
end
End
```

```

%%
function out=controlla_riga(vet_riga)
if vet_riga==1
    out=1;
elseif vet_riga==2
    out=2;
else
    out=0;
end

```

```

function visualizzaTris(M)

```

```

LN_WDT = 5;
MRK_SZ = 40
figure(1)
hold on
for ii = 1 : 3
    for jj = 1 : 3
        if(M(ii, jj) == 1)
            plot(jj, ii, 'bx', 'LineWidth', LN_WDT, 'MarkerSize', MRK_SZ)
        elseif(M(ii, jj) == 2)
            plot(jj, ii, 'ro', 'LineWidth', LN_WDT, 'MarkerSize', MRK_SZ)
        end
    end
end
end
% linee verticali
plot([1.5, 1.5], [0.5, 3.5], 'k-')
plot([2.5, 2.5], [0.5, 3.5], 'k-')
% linee orizzontali
plot([0.5, 3.5], [1.5, 1.5], 'k-')
plot([0.5, 3.5], [2.5, 2.5], 'k-')
hold off

```

### Domanda 3. Interpretazione e modifica del codice

```
clear
anno=2019;
stringa='Unità_Italia_17_Marzo_1861';
stringa2='';
for x=1:length(stringa)
    if stringa(x)>=48 && stringa(x)<=57
        stringa2=[stringa2 stringa(x)];
    end
end
anno_ui = str2double(stringa2(3:6));
n_anni_italia = anno-anno_ui;
disp(stringa2) %0
disp(['n_anni_italia = ', n_anni_italia]) %1
disp(['n_anni_italia = ', num2str(n_anni_italia)]) %2
whos %3
```

HINT: nella tabella ASCII lo '0' corrisponde al 48.

HINT: *str2double* converte una sequenza di caratteri passata in ingresso in un numero

HINT: *whos* stampa a schermo una tabella con le seguenti informazioni per ogni variabile

Name	Size	Bytes	Class
------	------	-------	-------

- Si dica se l'istruzione %1 è corretta. Perché?
- Si dica se l'istruzione %2 è corretta. Perché?
- Si riporti quello che viene stampato a schermo durante l'esecuzione del programma, considerando solo le istruzioni ritenute corrette.

Name	Size	Bytes	Class
------	------	-------	-------

**Domanda 4. Teoria (CP2, Tabella di Verità)**

- (a) Si costruisca la tabella di verità della seguente espressione booleana, badando alla precedenza tra gli operatori logici. (1 punto).

**not ( ( not ( A and not B ) ) or B or ( C and A ) )**

-

- (b) Si stabilisca il minimo numero di bit sufficiente a rappresentare in complemento a due i numeri  $A = 111_{\text{dec}}$  e  $B = -81_{\text{dec}}$ , li si converta, se ne calcolino la somma (A+B) e la differenza (A-B) in complemento a due e si indichi se si genera riporto sulla colonna dei bit più significativi e se si verifica overflow (1 punto).

## SOLUZIONE

$$\sim\{[\sim(A \& \sim B)] \parallel B \parallel (C \& A)\}$$

A	B	C	$\sim B$	A & $\sim B$	$\sim(A \& \sim B)$	$\sim(A \& \sim B) \parallel B$	C & A	$[\sim(A \& \sim B) \parallel B] \parallel (C \& A)$	$\sim\{[\sim(A \& \sim B) \parallel B] \parallel (C \& A)\}$
0	0	0	1	0	1	1	0	1	0
0	0	1	1	0	1	1	0	1	0
0	1	0	0	0	1	1	0	1	0
0	1	1	0	0	1	1	0	1	0
1	0	0	1	1	0	0	0	0	1
1	0	1	1	1	0	0	1	1	0
1	1	0	0	0	1	1	0	1	0
1	1	1	0	0	1	1	1	1	0

$$111 = 01101111$$

$$-81 = 10101111$$

$$111 - 81 = [0](1)00011110$$

$$111 + 81 = [1](0)11000000$$