



Politecnico di Milano

Facoltà di Ingegneria Civile, Ambientale e Territoriale

**Informatica (ICA-LC) [083668] – Informatica B [079904]**

**Informatica B [060054]**

**Prof. P. Plebani**

**Allievi Ingegneria Civile e Ambientale**

**Il prova in itinere (27 Gennaio 2011)**

<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>5</b>	<b>6</b>	<b>TOT</b>
<b>Punteggio max</b>	<b>4</b>	<b>8</b>	<b>6</b>	<b>6</b>	<b>6</b>	<b>2</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**.

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. Tracing (4 punti).

Data la seguente funzione vet

```
function vet = shuffle(vet, vals)

    pos = min(find(vet >= vals));

    if length(pos)>0
        if pos < length(vet)
            t = vet(1 : pos - 1);
            u = vet(pos : end);
            vet = [t(end : -1 : 1) , t , u(end : -1 : 1) , u];
        else
            vet = (find((vet > pos) & vet > 0));
        end
    else
        vet = vet(end : -1 : 1);
    end
```

indicare il valore di a e b mostrato a video ottenuto dall'esecuzione del seguente codice indicando i passaggi salienti

```
a = shuffle([1 2 34 3 33 74] , 7);
disp(['a = ' num2str(a)]);

b = shuffle([1 5 22 4 60] , 40);
disp(['b = ' num2str(b)]);
```

```
a = 2    1    1    2    74    33    3    34    34    3    33    74
b = 3    5
```

## Domanda 2. Funzioni e MATLAB (8 punti).

Scrivere un programma per la gestione di un magazzino dove ogni prodotto nel magazzino è univocamente identificato da un barcode (un numero intero) e da una tipologia (un carattere). Il software di gestione associa ad ogni prodotto due numeri, il primo che indica il numero di pezzi in stock il secondo che indica il numero di pezzi ordinati.

Si ipotizzi che `barcode`, `tipo`, `stock`, `ordine` siano 4 vettori, già popolati, contenenti tutte le informazioni necessarie per la gestione del magazzino. (l'i-simo elemento di `stock` e di `ordine` rappresentano le quantità relative al prodotto a cui è associato l'i-simo elemento di `barcode`).

Ad esempio:

```
barcodes = [123 ; 1312 ; 12312 ; 1231 ; 99123];
tipo = ['A' ; 'A' ; 'X' ; 'W' ; 'W'];
stock = [0 ; 300 ; 5 ; 6 ; 0];
ordine = [23 ; 100 ; 2 ; 100 ; 0];
```

Si scriva:

- la funzione `ricerca` che prende in ingresso un barcode e restituisce un messaggio contenente il tipo di prodotto, il numero di pezzi in stock ed in ordine.
- un esempio di chiamata alla funzione `ricerca`.
- la funzione `ricercaMancanti` che, a seconda di un parametro `P`, restituisce al programma chiamante
  - tutti i prodotti non presenti in stock ma in ordine (se `P = 0`)
  - tutti i prodotti non presenti in stock che non sono nemmeno in ordine (se `P = 1`)
  - tutti i prodotti per cui ci sono più pezzi in ordine che attualmente in stock (se `P = 2`)
- Scrivere un esempio di chiamata alla funzione `ricercaMancanti`
- Si scriva la funzione `aggiungiProdotto`, che permette di aggiungere al magazzino un nuovo prodotto (`barcode + stock + ordine`)
- Scrivere un esempio di chiamata alla funzione `aggiungiProdotto`

```
function ricerca(barcodes , tipo , stock , ordine , bc_to_find)
    bc_indexes = find(barcodes == bc_to_find);
    str = ['il prodotto corrispondente al barcode ' ,
          num2str(barcodes(bc_indexes(1)))];
    str = [str , ' È di tipo ' , tipo(bc_indexes(1))];
    str = [str , ' elementi in stock: ' , num2str(stock(bc_indexes(1))) ,
          ', in ordine: ' , num2str(ordine(bc_indexes(1)))];
    disp(str);
end

ricerca(barcodes , tipo , stock , ordine , 123)

function ricercaMancanti(barcodes , tipo , stock , ordine , P)

    switch P
        case 0
            bc_indexes = find((stock == 0) & (ordine > 0));
            str = ['prodotti non presenti in stock ma in ordine'];
        case 1
            bc_indexes = find((stock == 0) & (ordine == 0));
            str = ['prodotti non presenti in stock e non in ordine '];
        case 2
            bc_indexes = find(ordine > stock);
            str = ['prodotti di cui abbiamo più pezzi in ordine che in stock'];
    end

    disp (str)
```

```

for ii = 1 : length(bc_indexes)
    str = ['il prodotto corrispondente al barcode ' ,
          num2str(barcodes(bc_indexes(ii)))];
    str = [str , ' È di tipo ' , tipo(bc_indexes(ii))];
    str = [str , ' elementi in stock: ' , num2str(stock(bc_indexes(ii))) ,
          ', in ordine: ' , num2str(ordine(bc_indexes(ii)))];
    disp(str);
end

end

ricercaMancanti(barcodes , tipo , stock , ordine , 2)

function [barcodes , tipo , stock , ordine] = aggiungiProdotto(barcodes ,
tipo , stock , ordine , bc_toAdd , tipo_toAdd , stock_toAdd ,
ordine_toAdd)

    barcodes = [barcodes ; bc_toAdd];
    tipo = [tipo ; tipo_toAdd];
    stock = [stock ; stock_toAdd];
    ordine = [ordine ; ordine_toAdd];
end

[barcodes , tipo , stock , ordine] = aggiungiProdotto(barcodes , tipo ,
stock , ordine , 12312 , 'L', 12, 0)

```

### Domanda 3. Modello relazionale (6 punti).

Si definisca lo schema relazionale in grado di memorizzare le informazioni utili alla gestione di un palinsesto televisivo. Il palinsesto è composto da una serie di programmi ognuno identificato da un codice, da un nome, dal giorno e dall'ora di trasmissione e dalla durata. Sono anche memorizzati i presentatori televisivi (matricola, cognome, nome e programma condotto e il cachet per la presentazione).

Oltre agli schemi relazionali si indichino anche gli opportuni vincoli di chiave esterna e le chiavi primarie

Sulla base dello schema relazionale ottenuto, si definiscano in algebra relazionale le seguenti interrogazioni

- Elenco dei programmi trasmessi la domenica in cui il presentatore è pagato meno di 10000 euro.
- Elenco dei presentatori televisivi che presentano nella fascia oraria 10-12

PROGRAMMA (Codice, Nome, Giorno, Ora, Durata)



PRESENTATORE (Matricola, Cognome, Nome, Programma, Cachet)

$\pi_{\text{Codice, Programma, Nome}} (\sigma_{\text{cachet} < 10000 \wedge \text{giorno} = \text{'Domenica'}} \text{Programma} \bowtie_{\text{programma=codice}} \text{Presentatore})$

$\pi_{\text{Presentatore, Nome}} (\sigma_{\text{ora} > 10 \wedge \text{ora} < 12} \text{Programma} \bowtie_{\text{programma=codice}} \text{Presentatore})$

#### Domanda 4. SQL (6 punti)

Dato il seguente schema relazionale

Cantina (Nome, Indirizzo, Vitigno, ~~NomeProp~~, ~~CognomeProp~~)

Proprietario(Nome, Cognome, Indirizzo, Telefono)

Vino(Nome, Produzione, Anno, Cantina)

Si descrivano le espressioni SQL in grado di eseguire le seguenti interrogazioni

- Elenco dei vini prodotti nel 2000 presso la cantina 'VinoBuono'
- Quantità di vino prodotto nelle cantine lecchesi nel 2000
- Quantità di vino prodotto nelle cantine di "Mario" "Rossi"

```
SELECT Nome
FROM Vino
WHERE Cantina = 'VinoBuono' AND Anno = 2000
```

```
SELECT SUM(Produzione)
FROM Vino JOIN Cantina ON Vino.Cantina=Cantina.Nome
WHERE Vino.Anno=2000 AND Cantina.Indirizzo='Lecco'
```

```
SELECT SUM(Produzione)
FROM Vino JOIN Cantina ON Vino.Cantina=Cantina.Nome
WHERE Cantina.CognomeProp='Baracchi' AND Cantina.NomeProp = 'Mario'
```

**Domanda 5. Sistemi operativi (6 punti)**

Descrivere le caratteristiche, i vantaggi e gli svantaggi della rilocalizzazione statica e della rilocalizzazione dinamica. Inoltre indicare il ruolo della MMU in questo ambito

**Domanda 6. Domanda extra (2 punti).**

Descrivere il rapporto che intercorre tra il context switching dei processi e l'architettura di Von Neumann