

**PROVA SCRITTA DEL CORSO DI  
FONDAMENTI DI INFORMATICA  
CORSI DI LAUREA IN INGEGNERIA CHIMICA ED INGEGNERIA MECCANICA  
9 Settembre 2015**

**MOTIVARE IN MANIERA CHIARA LE SOLUZIONI PROPOSTE A CIASCUNO DEGLI ESERCIZI SVOLTI**

**ESERCIZIO 1 (7 punti)**

- (1) (3 punti) Indicare a quale legge dell'algebra booleana corrispondono le seguenti equivalenze, e cosa significano. Date due variabili booleane A e B, si dimostra che:  $\overline{A} \cdot \overline{B} = \overline{A + B}$  e che:  $\overline{A + B} = \overline{A} \cdot \overline{B}$ .
- (2) (4 punti) Applicare la legge del punto precedente alla seguente espressione:  $F = \overline{Z + X \cdot Y}$ , dove X, Y, Z, sono appunto variabili booleane. Verificare l'esattezza dell'applicazione mediante le tabelle di verità.

**ESERCIZIO 2 (3 punti)**

**Spiegare** in modo chiaro e sintetico a quale delle unità di un moderno calcolatore elettronico sono associati i seguenti compiti:

- 1) Esecuzione di operazioni aritmetiche
- 2) Trasmissione di un indirizzo da CPU a memoria
- 3) Decodifica delle istruzioni di macchina

**ESERCIZIO 3 (5 punti)**

<p>Relation Studenti</p> <p>(Matricola: char(5) PRIMARY KEY NOT NULL, Nome: char(20) NOT NULL, Data-N: date, Anno-Corso: integer, Corso-Laurea: char(1));</p> <p>Relation Corsi</p> <p>(Cod-Corso: char(6) PRIMARY KEY NOT NULL, Titolo: char(50), Docente: char(20));</p> <p>Relation Esami</p> <p>(Cod-Corso: char(5) NOT NULL, Matricola: char(5) NOT NULL, Data-E: date NOT NULL, Voto: integer NOT NULL, PRIMARY KEY (Cod-Corso, Matricola));</p>	<p>Facendo riferimento allo schema logico a sinistra:</p> <p>(2 punti) Creare in SQL la tabella Studenti.</p> <p>(3 punti) Scrivere in SQL la seguente interrogazione, precisandone il tipo: Estrarre nome e matricola degli studenti che hanno sostenuto il corso "Fondamenti di Informatica" prendendo un voto superiore a 25.</p>
--	--

**ESERCIZIO 4 (18 punti)**

1. (2 punti) Scrivere uno script Python che, ricevendo da tastiera un valore intero come stringa, lo converta in un valore in virgola mobile e lo stampi a video.
2. (3 punti) Scrivere un frammento di codice Python che, a partire dal seguente dizionario:  
`contatti={"Roli": 3294328, "Marcialis": 4930234, "Armano": 4783982}`  
stampi a video i valori relativi a ciascuna chiave.
3. (5 punti) Scrivere una funzione Python `leggi_valori` che ricevendo in ingresso un dizionario D ed una lista di chiavi L, restituisca una lista contenente la coppia chiave-valore per quelle chiavi presenti in D.  
Esempio: se `D=contatti` (v. esercizio 2), ed `L=["Marcialis", "Fumera"]`, la funzione restituirà la lista `[["Marcialis", 4930234]]` perché "Fumera" non è tra le chiavi di D.
4. (8 punti) Scrivere uno script Python che, a partire dal file "dati.txt", legga per ogni riga del file una lista di chiavi e salvi su file "output.txt" i valori indicati dalle chiavi di un dato dizionario D nella lista letta, separati ciascuno dal carattere di spaziatura, e poi vada a capo. Si usi la funzione sviluppata nell'esercizio precedente.

## Soluzioni

### ESERCIZIO 1

Si chiamano Leggi di De Morgan, e spiegano che con gli operatori AND/NOT o OR/NOT si può descrivere qualunque espressione booleana.

Se prendiamo l'espressione  $F = \overline{Z + \overline{X} \cdot \overline{Y}}$ , è possibile ad esempio applicare la prima equivalenza all'interno dell'espressione ottenendo:

$$F = \overline{Z + \overline{X + Y}}$$

Come si nota la stessa espressione di prima è ora descritta esclusivamente da operatori OR e NOT. Oppure possiamo applicare la seconda equivalenza all'espressione esterna:

$$F = \overline{Z} \cdot \overline{\overline{X} \cdot \overline{Y}}$$

Ottenendo un'espressione descritta esclusivamente da AND e NOT. Infine, applicando entrambe le equivalenze otteniamo l'espressione:

$$F = \overline{Z + \overline{X} \cdot \overline{Y}} = \overline{Z} \cdot \overline{\overline{X} \cdot \overline{Y}} = \overline{Z} \cdot \overline{\overline{X + Y}} = \overline{Z} \cdot (X + Y)$$

La verifica tramite tabelle di verità si lascia allo studente, che dovrà valutare se per ogni configurazione delle variabili X, Y, Z si ottiene la stessa espressione di F prima e dopo l'applicazione delle leggi.

### ESERCIZIO 2

Vedere dispense del corso.

### ESERCIZIO 3

```
(1)
CREATE TABLE STUDENTI
(MATRICOLA:   char(5),   PRIMARY KEY NOT NULL,
NOME:        char(20),   NOT NULL,
DATA-N:      date,
ANNO-CORSO:  integer,
CORSO-LAUREA: char(1));
```

```
(2)
SELECT NOME, STUDENTI.MATRICOLA
FROM STUDENTI, CORSI, ESAMI
WHERE STUDENTI.MATRICOLA=ESAMI.MATRICOLA
AND CORSI.COD-CORSO=ESAMI.COD-CORSO
AND TITOLO="FONDAMENTI DI INFORMATICA"
AND VOTO>25
```

Si tratta senza dubbio di una join, perché richiede un sottoinsieme del prodotto cartesiano delle relazioni `Studenti`, `Corsi` ed `Esami`.

#### ESERCIZIO 4

1. (2 punti) Scrivere uno script Python che, ricevendo da tastiera un valore intero come stringa, lo converta in un valore in virgola mobile e lo stampi a video.

```
x=raw_input()
x=float(x)
print x
```

2. (3 punti) Scrivere un frammento di codice Python che, a partire dal seguente dizionario:  
contatti={"Roli": 3294328, "Marcialis": 4930234, "Armano": 4783982}  
stampi a video i valori relativi a ciascuna chiave.

```
for key in contatti:
    print contatti[key]
```

3. (5 punti) Scrivere una funzione Python leggi\_valori che ricevendo in ingresso un dizionario D ed una lista di chiavi L, restituisca una lista contenente la coppia chiave-valore per quelle chiavi presenti in D.

Esempio: se D=contatti (v. esercizio 2), ed L=["Marcialis", "Fumera"], la funzione restituirà la lista [{"Marcialis", 4930234}] perché "Fumera" non è tra le chiavi di D.

```
def leggi_valori(D,L):
    coppie=[]
    for k in L:
        if k in D:
            coppie=coppie+[[k,D[k]]]
    return coppie
```

4. (8 punti) Scrivere uno script Python che, a partire dal file "dati.txt", legga per ogni riga del file una lista di chiavi e salvi su file "output.txt" i valori indicati dalle chiavi di un dato dizionario D nella lista letta, separati ciascuno dal carattere di spaziatura, e poi vada a capo. Si usi la funzione sviluppata nell'esercizio precedente.

```
infile=open("dati.txt","r")
outfile=open("output.txt","w")

line=infile.readline()
while line!="":
    chiavi=line.split()
    coppie=leggi_valori(D,chiavi)
    outline=""
    for c in coppie:
        outline=outline+str(c[1])+" "
    outfile.write(outline+"\n")
    line=infile.readline()

infile.close()
outfile.close()
```