

**PROVA SCRITTA DEL CORSO DI
FONDAMENTI DI INFORMATICA 1
CORSI DI LAUREA IN INGEGNERIA CHIMICA ED INGEGNERIA MECCANICA
15/9/2016**

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

NOME: _____ **COGNOME:** _____ **MATRICOLA:** _____

ESERCIZIO 1 (5 punti)

- 1) (2 punti) Scrivere la tabella di verità dell'espressione booleana dell'espressione $\bar{A} + A \cdot B$ e semplificare l'espressione qualora sia possibile.
- 2) (3 punti) Calcolare il complemento a due della stringa a quattro bit 1001 e scrivere il corrispondente valore decimale.

ESERCIZIO 2 (5 punti)

Descrivere in modo chiaro e sintetico la pila ISO/OSI nelle reti di calcolatori.

ESERCIZIO 3 (5 punti)

1. (2 punti) Cosa rappresenta la figura a fianco?
2. (3 punti) Scrivere in SQL la query: "Estrarre il nome degli studenti che hanno dato esami con prof. Marcialis".

Relation Studenti

(Matricola: char(5) PRIMARY KEY NOT NULL,
Nome: char(20) NOT NULL,
Data-N: date,
Anno-Corso: integer,
Corso-Laurea: char (1));

Relation Corsi

(Cod-Corso: char(6) PRIMARY KEY NOT NULL,
Titolo: char(50),
Docente: char(20));

Relation Esami

(Cod-Corso: char(6) NOT NULL,
Matricola: char(5) NOT NULL,
Data-E: date NOT NULL,
Voto: integer NOT NULL,
PRIMARY KEY (Cod-Corso, Matricola));

ESERCIZIO 4 (18 punti)

Si vuole implementare la relazione Esami dell'Esercizio 3 attraverso un dizionario Python, chiamato appunto `Esami`, in cui i vari nomi di attributo corrispondono ad altrettante chiavi. I tipi per ciascun attributo sono ridefiniti come segue: `cod-corso`, `matricola`, `voto` sono interi, `data` è una stringa secondo il formato "gg.mm.aaaa".

Sulla base di questo, implementare le seguenti funzioni:

1) (4 punti) Funzione `popola(nomeFile)`: legge da file, il cui nome è passato come parametro, i dati relativi ad un certo numero di esami i cui valori di ciascun attributo sono separati da spazi. Tali valori vanno inseriti in un dizionario restituito in uscita.

2) (5 punti) Funzione `proietta(dizionario, lista_attributi)`, stampa a video tutti i valori di attributo di ciascuna tupla, elencati nel parametro `lista_attributi`. Ogni tupla sarà separata dall'altra mediante carattere di "a capo".

3) (6 punti) Funzione `seleziona(dizionario, attributo, condizione)`, che stampa a video i valori relativi ad `attributo` delle tuple soddisfacenti ciascuno una condizione presente in `condizione`. Quest'ultima è una stringa strutturata così:

"<operatore_di_condizione> <valore>"

Dove <operatore_di_condizione> equivale a uno di tre possibili operatori di confronto `=`, `>`, `<`. <valore> è invece il relativo valore dell'attributo rispetto al quale va applicato l'operatore di confronto.

Ad esempio la chiamata a funzione seguente:

```
seleziona(Esami, "Data", "> 25.01.2012")
```

stamperà a video tutte le date d'esame conseguito dopo il 25.01.2012.

Utilizzando le predette funzioni, si scriva infine la sequenza di istruzioni Python che dopo avere popolato la relazione Esami da file "esami.txt", stampi a video tutti i valori di tupla relativi agli attributi `Matricola` e `Data`, e poi stampi a video tutti gli esami con voti maggiori di 25. (3 punti)

ESERCIZIO 1 (5 punti)

1.

A	B	AB	A'+AB
0	0	0	1
0	1	0	1
1	0	0	0
1	1	1	1

Come si vede $\bar{A} + AB = \overline{A\bar{B}} = \bar{A} + B$. L'espressione semplificata è stata ottenuta osservando che l'espressione è nulla solo in una occasione, corrispondente ad $A\bar{B}$ (riga in rosso della tabella, con A=1 e B=0). Negando questa espressione si ottiene, per la legge di De Morgan, l'espressione semplificata cercata.

2.

Il complemento a due di 1001 si ottiene negando tutti i bit e poi aggiungendo 1: $0110+1=0111$. Tale valore corrisponde evidentemente a +7 in base al seguente calcolo: $0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 0 + 4 + 2 + 1 = 7$.

ESERCIZIO 2 (5 punti)

V. dispense del corso.

ESERCIZIO 3 (5 punti)

1. Si tratta di uno schema di relazione, che ha il compito di descrivere la struttura base della relazione stessa, con gli attributi ed i relativi vincoli e tipi.
- 2.

```
SELECT NOME
FROM ESAMI, CORSI, STUDENTI
WHERE STUDENTI.MATRICOLA=ESAMI.MATRICOLA
AND CORSI.COD-CORSO=ESAMI.COD-CORSO
AND DOCENTE="MARCIALIS"
```

ESERCIZIO 4 (18 punti)

```
def popola(nomeFile):
    idf=open(nomeFile,"r")
    d={"Cod-corso":[], "Matricola":[], "Data":[], "Voto":[]}

    linea=idf.readline()
    while linea!="":
        valori=linea.split()
        d["Cod-corso"]= d["Cod-corso"]+[int(valori[0])]
        d["Matricola"]= d["Matricola"]+[int(valori[1])]
        d["Data"]= d["Data"]+[valori[2]]
        d["Voto"]= d["Voto"]+[int(valori[3])]
        linea=idf.readline()

    idf.close()
    return d
```

```

def proietta(esami, attributi):
    t=""
    ntuple=len(esami["Cod-corso"])
    i=0
    while i<ntuple:
        for k in attributi:
            t=t+str(esami[k][i])+" "
            t=t+"\n"
            i=i+1
    print t

def seleziona(esami, attributo, opcond):
    valori=esami[attributo]

    op=opcond.split()
    if attributo!="Data":
        op[1]=int(op[1])

    for v in valori:
        if op[0]==">":
            if v>op[1]:
                print str(v)
        elif op[0]=="<":
            if v<op[1]:
                print str(v)
        else:
            if v==op[1]:
                print str(v)

Esami=popola("esami.txt")
proietta(Esami, ["Matricola", "Data"])
seleziona(Esami, "Voto", "> 25")

```

NOTA: sebbene la soluzione presentata presenti un errore in quanto il formato della stringa fornito per il campo Data, "gg.mm.aaaa", non permette un confronto diretto tra due date, essa verrà considerata accettabile in fase di correzione. Il formato corretto è "aaaa.mm.gg".