

**PROVA SCRITTA DEL CORSO DI
FONDAMENTI DI INFORMATICA 1
CORSI DI LAUREA IN INGEGNERIA CHIMICA ED INGEGNERIA MECCANICA
20/6/2016**

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

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

ESERCIZIO 1 (5 punti)

1. (3 punti) Si consideri la rappresentazione in eccesso 36 su un campo di sei bit per la rappresentazione di numeri interi. Si indichi a quale valore decimale corrisponde la stringa 010010 rappresentata con tale eccesso.
2. (2 punti) Scrivere, utilizzando gli operatori fondamentali dell'algebra booleana (AND, OR, NOT), l'espressione che verifica la condizione binaria $A > B$ (si legge "A strettamente maggiore di B"), essendo A e B due bit da confrontare.

ESERCIZIO 2 (5 punti)

1. (3 punti) Spiegare in modo chiaro e sintetico la struttura di un pacchetto.
2. (2 punti) Quale livello ISO/OSI si occupa della composizione e decomposizione dei pacchetti da inviare/ricevere?

ESERCIZIO 3 (5 punti)

1. (2 punti) Discutere in modo chiaro e sintetico cosa differenzia l'uso di un DBMS che gestisca archivi eterogenei da un accesso software dedicato ad ogni archivio.
2. (3 punti) Siano date le relazioni popolate descritte in figura. Sapendo che Matricola è chiave primaria delle relazioni Docenti e Allievi, scrivere la seguente interrogazione: selezionare i nomi e cognomi dei professori che hanno almeno due corsi ed almeno un allievo con voto maggiore od uguale a 20. La risposta alla query non deve presentare ripetizioni.

Docenti

Nome_D	Cognome_D	Matricola	Corsi_tenuti
Gian Luca	Marcialis	1	2
Luca	Didaci	2	4
Michele	Marchesi	3	5
Giuliano	Armano	4	3
Giorgio	Fumera	5	2
Fabio	Roli	6	5

Allievi

Matricola	Nome	Cognome	Max_voto	Docente
1	Giancarla	Ferrai	30	3
2	Marco	Pistis	30	2
3	Maria	Desogus	18	1
4	Silvia	Ferrai	30	1
5	Gianguido	Pistis	25	2

ESERCIZIO 4 (18 punti)

Dovendo sostenere a breve l'esame di Geometria, alcuni vostri colleghi vi chiedono una mano per implementare un insieme di funzioni Python (una libreria per intenderci) in grado di leggere delle matrici di dimensione qualunque da file, effettuare somma o trasposizione, e salvare in un altro file il relativo risultato. I due operatori devono anche restituire un messaggio di errore nel caso in cui le dimensioni delle due matrici da elaborare risultino incompatibili per la somma.

Innanzitutto decidete di usare, per rappresentare le matrici, una lista di liste. Ad esempio, la lista `[[1, -3], [2, 4]]` rappresenta la matrice: $\begin{bmatrix} 1 & -3 \\ 2 & 4 \end{bmatrix}$.

Una volta definita la struttura dati di cui sopra, decidete che il modo migliore di soddisfare i vostri colleghi è implementare le seguenti funzioni:

- 1) (3 punti) Funzione `leggi(nomeFile)`: legge da file il cui nome è passato come parametro una matrice e la restituisce. Il file è formattato per riga con i valori floating point rappresentanti ciascun elemento di una data riga, separati da spazio. Il carattere "a capo" annuncia l'eventuale presenza della prossima riga.
- 2) (1 punto) Funzione `dimensioni(matrice)`: restituisce il numero di righe e il numero di colonne della matrice fornita in ingresso.
- 3) (4 punti) Funzione `somma(matrice1, matrice2)`: restituisce una nuova matrice somma delle due fornite in ingresso. Il valore restituito è *False* se le due matrici hanno dimensioni incompatibili.
- 4) (5 punti) Funzione `trasponi(matrice)`: restituisce una nuova matrice trasposta rispetto a quella fornita in ingresso.
- 5) (2 punti) Funzione `salva(nomeFile, matrice)`: salva nel file il cui nome è fornito in ingresso la matrice, formattando il file come spiegato al punto 1.

Utilizzando le predette funzioni, si scriva infine la sequenza di istruzioni Python che legga due matrici presenti nei file "m1.txt" e "m2.txt" e salvi la loro somma su "s.txt", stampando a video il messaggio "Dimensioni incompatibili" nel caso non sia possibile calcolarla (3 punti).

ESERCIZIO 1

1. Applicando la definizione, rappresentare sei bit in eccesso 36 significa che la configurazione 000000 corrisponde al valore -36, ovvero che la configurazione 100110 corrisponde al valore 0. Si veda anche Slide Cap. 1, n. 31. Per ottenere il valore corrispondente a 010010, basta prima di tutto convertirlo in decimale ottenendo $010010 = 2^4 + 2^1 = 17$ e poi sottrargli l'eccesso, ottenendo il valore -19, ovvero il valore cercato.
2. E' sufficiente stilare la tabella di verità:

A	B	A>B
0	0	0
0	1	0
1	0	1
1	1	0

Da questa tabella si evince che l'espressione cercata è pari ad 1 in corrispondenza dell'AND di A in forma non negata con B in forma negata, ovvero $A \cdot \bar{B}$.

ESERCIZIO 2

Vedi le dispense del corso, Cap. 5, nn. 22, 28.

ESERCIZIO 3

1. Vedi le dispense del corso, Cap. 4, nn. 7-10.
- 2.

```
SELECT DISTINCT NOME_D, COGNOME_D
FROM DOCENTI, ALLIEVI
WHERE DOCENTI.MATRICOLA=DOCENTE
AND CORSI_TENUTI>=2
AND MAX_VOTO>=20
```

ESERCIZIO 4

```
def leggi(nomeFile): #il parametro passato è il nome del file
    idf=open(nomeFile,"r")
    riga=idf.readline()
    m=[]
    while riga!="": #finché riga non vuota, ovvero fine del file...
        rm=riga.split()
        lr=[]
        for x in rm:
            lr=lr+[float(x)] #convertito ogni singolo elemento in un numero
        m=m+[lr]
    idf.close()
    return m

def dimensioni(mat): #il parametro è la matrice
    n=len(mat) #numero righe
    m=len(mat[0]) #numero colonne
    return n, m
```

```

def somma(mat1,mat2): #due matrici come parametri
    d1=dimensioni(mat1)
    d2=dimensioni(mat2)
    if d1!=d2:          #il != va bene anche per le liste
        return False    #dimensioni incompatibili

    n=d1[0]
    m=d1[1]
    i=0
    s=[]
    while i<n:
        j=0
        rs=[]
        while j<m:
            rs=rs+[mat1[i][j]+mat2[i][j]]
            j=j+1
        s=s+[rs]
        i=i+1

    return s

def trasponi(mat):
    d=dimensioni(mat)

    r=[0] * d[0] #righe e colonne vanno invertite
    t=[r] * d[1]

    i=0
    while i<d[0]:
        j=0
        while j<d[1]:
            t[j][i]=mat[i][j]
            j=j+1
        i=i+1
    return t

def salva(nomeFile,matrice):
    idf=open(nomeFile,"w")
    for riga in matrice:
        for elemento in riga:
            idf.write(str(elemento)+ " ")
        idf.write("\n")
    idf.close()

#Programma principale
m1=leggi("m1.txt")
m2=leggi("m2.txt")
m=somma(m1,m2)
if not m:
    print "Dimensioni incompatibili"
else:
    salva("s.txt",m)

```