

**PROVA SCRITTA DEL CORSO DI  
FONDAMENTI DI INFORMATICA 1  
CORSI DI LAUREA IN INGEGNERIA CHIMICA ED INGEGNERIA MECCANICA  
4/7/2016**

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

**NOME:** \_\_\_\_\_ **COGNOME:** \_\_\_\_\_ **MATRICOLA:** \_\_\_\_\_

**ESERCIZIO 1 (5 punti)**

1. (3 punti) Un'immagine a livelli di grigio compresi tra 0 e 255, estremi inclusi, ha dimensione 640x480 pixel. Quanti Kbyte occupa l'immagine? Cosa significano i valori 0 e 255?
2. (2 punti) Semplificare l'espressione  $\bar{A}B + A + \bar{B}$  con le leggi dell'algebra booleana.

**ESERCIZIO 2 (5 punti)**

Si hanno a disposizione 8 bit per rappresentare le istruzioni in un calcolatore elettronico (si veda lo schema in figura). Supponendo di suddividere questi bit nei campi codice operativo ed operando:

1. (3 punti) Quanti bit, sugli 8 disponibili, è necessario dedicare al campo operando affinché un'istruzione possa indirizzare 64 parole di memoria?
2. (2 punti) Con la scelta relativa al punto 1, è possibile realizzare le istruzioni LOAD, STORE, ADD, SUB, MUL, JUMP con i bit rimanenti per il codice operativo?

B7	B6	B5	B4	B3	B2	B1	B0

In Figura gli 8 bit a disposizione da suddividere nei campi codice operativo ed operando secondo le specifiche dell'esercizio.

**ESERCIZIO 3 (5 punti)**

1. (2 punti) Discutere in modo chiaro e sintetico cosa significa DML e DDL.
2. (3 punti) Dato lo schema logico in figura, scrivere la query: "Estrarre matricola, nome e voto degli studenti che hanno sostenuto l'esame di un corso insegnato da Didaci o 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)

La competizione tra ragazzi e ragazze è diventata fortissima dal presente anno accademico. Per valutare il livello di bravura all'esame di Fondamenti di Informatica, gli studenti decidono di implementare in Python un programma capace di leggere un file contenente la matricola, il sesso (indicato con il carattere "M" o "F") e il voto conseguito da ciascuno e trarne delle informazioni statistiche per valutare chi siano i più bravi in media tra maschi e femmine, e chi sia il migliore tra i ragazzi e chi tra le ragazze.

Per questo scopo, si decide di memorizzare le tre informazioni per studente scritte sopra in una lista, e memorizzare tutti i dati letti in una lista di liste.

Si implementano poi le seguenti funzioni:

- 1) (3 punti) Funzione `leggi(nomeFile)`: legge da file, il cui nome è passato come parametro, i dati relativi agli studenti formattati per riga (ovvero ogni riga contiene la tripla matricola-sesso-voto separati da spazio) e restituisce una lista di liste secondo la specifica di cui sopra.
- 2) (4 punti) Funzione `Estrai_miglior_studente(sesso, listaDiListe)`, che ricevendo in ingresso una lista di liste organizzata secondo specifica, restituisce la matricola ed il voto della persona che ha voto più alto entro gli studenti del sesso indicato come parametro.
- 3) (5 punti) Funzione `Calcola_media(sesso, listaDiListe)`, che ricevendo in ingresso una lista di liste organizzata secondo specifica, restituisce la media voti relativa al sesso introdotto come parametro.
- 4) (3 punti) Funzione `Scrivi_risultati(lista)` che ricevendo in ingresso una lista contenente nell'ordine la matricola della migliore studentessa ed il relativo voto, la matricola del miglior studente ed il relativo voto, la media delle ragazze e la media dei ragazzi, scriva queste informazioni su file "risultati.txt" che risulterà così formattato:

```
<matricola studentessa>    <voto>
<matricola studente>      <voto>
<media studentesse>
<media studenti>
```

Utilizzando le predette funzioni, si scriva infine la sequenza di istruzioni Python che legga le informazioni dal "studenti.txt" e salvi le relative statistiche estratte su file "risultati.txt" (3 punti).

### ESERCIZIO 1 (5 punti)

1. Per rappresentare i 256 livelli di grigio considerati occorrono 8 bit, ovvero 1 byte per pixel (un pixel equivale ad un possibile valore di grigio per un dato punto dell'immagine). Un'immagine delle dimensioni date occupa  $640 \times 480 = 307200$  byte, ovvero 300 Kbyte, che si ottengono dividendo le dimensioni in byte per 1024 (1 Kbyte = 1024 byte).
2. E' sufficiente applicare la legge di De Morgan per osservare che:

$$\bar{A}B + A + \bar{B} = \overline{\bar{A}B} + A + \bar{B} = \overline{A + \bar{B}} + A + \bar{B} = 1$$

### ESERCIZIO 2 (5 punti)

1. Dovendo indirizzare  $64=2^6$  parole, sono necessari **6** bit per il campo operando, ottenendo lo schema in figura (C.O. sta per codice operativo). Come si vede, gli ultimi sei bit sono dedicati al campo operando:

B7	B6	B5	B4	B3	B2	B1	B0
C.O.		Campo operando					

2. Rimanendo  $8-6=2$  bit per il codice operativo come da schema soprastante, sono implementabili al massimo  $2^2=4$  istruzioni. Essendo indicate ben sei istruzioni (LOAD, STORE, ADD, SUB, MUL, JUMP) esse non sono tutte realizzabili.

### ESERCIZIO 3

1. Vedi le dispense del corso, Cap. 4, n. 13.

2.

```
SELECT DISTINCT STUDENTI.MATRICOLA, NOME, VOTO
FROM STUDENTI, CORSI, ESAMI
WHERE STUDENTI.MATRICOLA=ESAMI.MATRICOLA
AND CORSI.COD-CORSO=ESAMI.COD-CORSO
AND (DOCENTE='MARCIALIS' OR DOCENTE='DIDACI')
```

## ESERCIZIO 4

```
def leggi(nomeFile):          #il parametro passato è il nome del file
    idf=open(nomeFile,"r")
    riga=idf.readline()
    lista=[]
    while riga!="":          #finché riga non vuota, ovvero fine del file
        rm=riga.split()
        lista=lista+[rm]
    idf.close()
    return lista

def estrai_miglior_studente(sesso,lista):
    massimo=0
    for el in lista:
        if el[1]==sesso:
            x=int(el[2])
            if x>massimo:
                massimo=x
                bestmatricola=el[0]
    return bestmatricola, massimo

def calcola_media(sesso, lista):
    media=0.0
    nsesto=0.0
    for el in lista:
        if el[1]==sesso:
            media=media+float(el[2])
            nsesto=nsesto+1.
    media=media/nsesto
    return media

def scrivi_risultati(lista):
    idf=open("risultati.txt","w")
    t=lista[0]
    idf.write(t[0]+" "+str(t[1])+"\n")
    t=lista[1]
    idf.write(t[0]+" "+str(t[1])+"\n")
    idf.write(str(lista[2])+"\n"+str(lista[3]))
    idf.close()

#Programma principale
dati=leggi("studenti.txt")
lista=[estrai_miglior_studente("F",dati)]
lista=lista+[estrai_miglior_studente("M",dati)]
lista=lista+[calcola_media("F",dati)]
lista=lista+[calcola_media("M",dati)]
scrivi_risultati(lista)
```