

**PROVA SCRITTA DEL CORSO DI
FONDAMENTI DI INFORMATICA
CORSI DI LAUREA IN INGEGNERIA CHIMICA ED INGEGNERIA MECCANICA
21 febbraio 2019**

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

ESERCIZIO 1 (4 punti)

Indicare il minimo numero di bit necessario per rappresentare il valore intero senza segno 4096.

ESERCIZIO 2 (4 punti)

Siano date le istanze di relazione:

Docenti

Nome_D	Cognome_D	Matricola	Corsi_tenuti
Gian Luca	Marcialis	1	2
Carlo	Mei	2	4
Luca	Didaci	3	5
Matteo	Fraschini	4	3
Giorgio	Fumera	5	2
Fabio	Roli	6	5

Allievi

Matricola	Nome	Cognome	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

(1 punto) Indicare la cardinalità di ognuna delle due relazioni.

(3 punti) Scrivere in SQL l'interrogazione: "Si trovi nome e cognome degli studenti che hanno conseguito un voto minore di 25 con docenti che hanno almeno 2 corsi".

ESERCIZIO 3 (25 punti)

(7 punti) Un vostro amico ha ideato un'app di sicuro successo: crea una rete di contatti reciprocamente interessati ad una relazione sentimentale. Chi si iscrive alla rete mediate questa app, attivandola, riceverà a fine giornata l'elenco di tutte le persone giunte in prossimità del vostro cellulare durante la giornata che sono interessate ad una relazione sentimentale, ovviamente a loro volta iscritte all'app. Il formato di questo elenco è un file "papabili.txt", per ogni riga del quale è riportato in sequenza il nome della persona, il colore dei capelli ("Biondo"/"Castano"/"Scuri"), l'età, l'altezza e l'indirizzo e-mail.

Tuttavia, dopo un primo periodo di collaudo, si scopre che gli elenchi giornalieri che vengono ricevuti sono lunghissimi (tante persone si sono iscritte!), così che risulta difficilissimo scegliere la persona da contattare. Così il vostro amico pensa che sia il caso di aggiungere una speciale opzione per la quale ognuno possa inserire il colore dei capelli, l'età minima e l'altezza massima della persona alla quale si potrebbe essere potenzialmente interessati. Il sistema farebbe quindi un censimento automatico andando a ridurre il numero di persone a partire dall'elenco presente in "papabili.txt" e scrivendo l'elenco ridotto in un nuovo file "papabilissimi.txt", dove però verrebbe scritto soltanto il nome e l'indirizzo e-mail nel caso venissero soddisfatti tutti e tre i criteri. Tutte le persone non soddisfacenti tutti i criteri verrebbero escluse. Infine, il sistema segnalerebbe all'iscritto il messaggio "Oggi ci sono <X> papabili per te" oppure "Oggi non ci sono papabili per te, mi spiace!", a fine elaborazione, su video, con <X> numero di papabili calcolati in base ai criteri.

Poiché però è a corto di programmatori, chiede a voi di scrivere il codice Python della nuovissima funzionalità, in modo che possa integrarla nella sua app innovativa. Accettate e, dopo aver negoziato il compenso, vi mettete a lavoro. Decidete che:

- L'inserimento degli elementi desiderati avvenga in un dizionario che chiamate appunto desiderata. Le chiavi del dizionario sono rispettivamente "ColoreDeiCapelli", "EtaMinima", "AltezzaMassima", la prima punterà ad un valore stringa, mentre la seconda e la terza ad intero.
- L'elenco letto da "papabili.txt" dovrà finire in una lista di liste, dove ciascuna sottolista è costituita dai valori letti per riga: il nome della persona, il colore dei capelli, l'età, l'altezza e l'indirizzo e-mail.

Decidete poi di implementare le seguenti funzioni:

- (3 punti) `inserisciDesiderata()`: legge da tastiera il colore dei capelli, l'età minima e l'altezza massima desiderate e restituisce un dizionario contenente i valori inseriti indicizzati dalle chiavi indicate sopra.
- (7 punti) `leggiElenco(nomefile)`: legge da file di nome `nomefile`, formattato come indicato sopra, ciascuna riga e la deposita in una lista di liste, restituita in uscita.
- (5 punti) `selezionaPapabile(lista, desiderata)`: riceve in ingresso una lista corrispondente ad una riga di valori letti dal file "papabili.txt" e restituisce `True` se tutti i criteri di colore di capelli, età minima e altezza massima presenti nel dizionario desiderata corrispondono a quelli presenti nella lista, `False` altrimenti.
- (3 punti) `scriviPapabilissimo(f, l)`: scrive su file `f`, già aperto in modalità append, i valori presenti nella lista `l` in un'unica riga. La lista `l` contiene solo stringhe.

Soluzione dell'esercizio 1.

Poiché $4096=2^{12}$, per rappresentare tale valore occorrono almeno 13 bit.

Soluzione dell'esercizio 2.

La cardinalità delle relazioni, dato dal numero delle tuple presenti, è pari rispettivamente a 6 ed a 5.

```
SELECT NOME, COGNOME
FROM ALLIEVI, DOCENTI
WHERE
    DOCENTI.MATRICOLA=DOCENTE
AND
    VOTO<=25
AND
    CORSI_TENUTI>=3
```

Soluzione dell'esercizio 3.

```
def inserisciDesiderata():
    capelli=raw_input("Dammi il colore dei capelli preferito
(biondo/castano/scuro):\n")
    eta=input("Dammi l\'eta\' minima desiderata:\n")
    altezza=input("Dammi l\'altezza massima desiderata:\n")

    return {"ColoreDeiCapelli":capelli, "EtaMinima": eta,
"AltezzaMassima":altezza}

def leggiElenco(nomefile):
    f=open(nomefile, "r")

    l=[]
    r=f.readline()
    while r!="":
        r=r.split()
        r[2]=int(r[2])
        r[3]=int(r[3])
        l=l+[r]
        r=f.readline()

    f.close()
    return l

def selezionaPapabile(candidato, desiderata):

    capelliOK=candidato[1]==desiderata["ColoreDeiCapelli"]
    etaOK=candidato[2]>=desiderata["EtaMinima"]
    altezzaOK=candidato[3]<=desiderata["AltezzaMassima"]

    return capelliOK and etaOK and altezzaOK

def scriviPapabile(f,lista):
    s=" ".join(lista)
    f.write(s+"\n")

#Programma principale
d=inserisciDesiderata()
l=leggiElenco("papabili.txt")
n=0
for papabile in l:
    p=selezionaPapabile(papabile,d)
    if p:
        f=open("papabilissimi.txt","a")
        scriviPapabile(f,[papabile[0],papabile[-1]])
        f.close()
        n=n+int(p)
    if n:
        print "Oggi ci sono "+str(n)+" papabili per te!\n"
    else:
        print "Oggi non ci sono papabili per te, mi spiace!\n"
```