

SIMULAZIONE DELLA PROVA SCRITTA DEL CORSO DI
FONDAMENTI DI INFORMATICA 1
CORSO DI LAUREA IN INGEGNERIA CHIMICA E INGEGNERIA MECCANICA
24 maggio 2017

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

ESERCIZIO 1 (4 punti)

Verificare l'identità $A' + AB + AC = A' + B + C$. Usare le proprietà e teoremi dell'algebra booleana oppure le tabelle di verità.

ESERCIZIO 2 (4 punti)

Siano date le relazioni popolate:

Docenti

Nome_D	Cognome_D	Matricola_Studente	Matricola_Docente
Gian Luca	Marcialis	1	1
Giulio	Concas	4	2
Giorgio	Fumera	4	3
Michele	Marchesi	3	4
Giorgio	Giacinto	5	5
Giuliano	Armano	2	6
Fabio	Roli	1	7

Allievi

Matricola	Nome	Cognome	Media
1	Giancarla	Perrai	30
2	Roborbio	Listis	30
3	Maria	Fesogus	18
4	Silvia	Dilvi	30
5	Giangiorgio	Pistillone	25

(2 punti) Precisare, motivando la risposta, il grado e la cardinalità di ciascuna delle due relazioni.

(2 punti) Scrivere in SQL la seguente interrogazione: estrarre nome e cognome dei docenti per i quali almeno uno studente ha preso un voto superiore a 25.

ESERCIZIO 3 (25 punti)

(6 punti) Scrivere un programma in linguaggio Python che legga da un file "input.txt" una sequenza di triple formata da una stringa e due valori interi, indicanti voto e crediti associati all'esame il cui nome è appunto la stringa, scriva su video l'espressione "L'esame il cui voto presenta il contributo più vicino alla media pesata è: ", seguito dal nome dell'esame. La stringa con la quale è indicato il nome dell'esame non presenta spazi ma al massimo caratteri "_" se composta da più parole.

Nota: la media pesata si calcola come $\frac{\sum_{i=1}^n voto_i \cdot peso_i}{\sum_{i=1}^n peso_i}$.

Esempio:

File "input.txt"		
Fondamenti_di_Informatica	30	6
Analisi_matematica_1	28	8
Chimica	27	5
Merendine applicate	20	5

Con l'esempio di sopra, si ha a video (la media pesata generata è 26.625):

L'esame il cui voto moltiplicato per i suoi crediti da contributo più vicino alla media pesata è Chimica.

Nello scrivere il programma si implementino seguenti funzioni.

(5 punti) Funzione `leggi_voti` con:

- parametri in ingresso: la stringa del nome del file
- parametri in uscita: Un dizionario la cui chiave è il nome dell'esame e i cui valori associati sono rappresentati da una lista con il voto e il numero dei crediti.

Legge da file `nomefile` una sequenza di coppie di interi e li memorizza nelle opportune strutture dati (liste/dizionari).

(3 punti) Funzione `somma` con:

- parametri in ingresso: una lista di valori interi
- parametri in uscita: la somma degli elementi della lista/del vettore

Somma gli elementi di una sequenza di numeri e la restituisce.

(5 punti) Funzione `calcola_minimo` con

- parametri in ingresso: la lista di interi e un valore di riferimento `x`
- parametri in uscita: il valore la cui distanza in valore assoluto da `x` è più piccola.

(3 punti) Funzione `estrai_voti` con:

- parametri in ingresso: il dizionario creato da `leggi_file`
- parametri in uscita: la lista dei voti per esame

(3 punti) Funzione `scrivi_risultato` con:

- parametro in ingresso: il dizionario letto da `leggi_file`, un valore di voto

Scriva su video la frase indicata nel testo con l'esame o i nomi degli esami i cui voti sono pari al voto inserito come parametro.

Soluzione dell'esercizio n.1

$$A' + AB + AC = A' + A * (B + C) = (A' + A) * (A' + B + C) = A' + B + C$$

Per la proprietà distributiva e dell'elemento complementare

Soluzione dell'esercizio n.2

1. Il grado corrisponde al numero di attributi (colonne) per relazione, la cardinalità al numero di tuple presenti (righe esclusa quella con la lista dei nomi degli attributi). Quindi entrambe le relazioni hanno grado 4, ma la cardinalità della prima è 7 mentre quella della seconda è 5.
2.

```
SELECT NOME_D, COGNOME_D
FROM DOCENTI, ALLIEVI
WHERE MATRICOLA=MATRICOLA_STUDENTE
AND VOTO>25
```

Soluzione dell'esercizio n.3

```
def leggi_voti(nomefile):
```

```
    f=open(nomefile,"r")
```

```
    d={}
```

```
    linea=f.readline()
```

```
    while linea!="":
```

```
        dati=linea.split()
```

```
        d[dati[0]]=[int(dati[1]), int(dati[2])]
```

```
        linea=f.readline()
```

```
    f.close()
```

```
    return d
```

```
def somma(v):
```

```
    s=0
```

```
    for x in v:
```

```
        s=s+x
```

```
    return s
```

```
def calcola_minimo(l,x):
```

```
    minabs=abs(l[0]-x)    #in luogo di abs si può creare una funzione apposita
```

```
    minv=l[0]
```

```
    for el in l[1:]:
```

```
        v=abs(el-x)
```

```
        if v<minabs:
```

```
            minabs=v
```

```
            minv=el
```

```
    return minv
```

```

def estrai_voti(d):
    l=[]
    for k in d:
        l=l+[d[k][0]]
    return l

def scrivi_risultato(d,mv):
    s=""
    for k in d:
        if d[k][0]==mv:
            s=s+k+" "
    print("Il voto il cui contributo alla media pesata è più vicino alla
media stessa e' " + s)

def main(): #non necessario inserire il corpo principale come funzione
    d=leggi_voti("input.txt")

    pesi=[]    #si può inserire questa parte in una funzione apposita
    media_pesata=0
    for k in d:
        media_pesata=media_pesata+d[k][0]*d[k][1]
        pesi=pesi+[d[k][1]]
    v=somma(pesi)
    media_pesata=float(media_pesata)/float(v)

    voti=estrai_voti(d)
    mvoto=calcola_minimo(voti,media_pesata)
    scrivi_risultato(d,mvoto)v

```