

SECONDA PROVA INTERMEDIA DEL CORSO DI
FONDAMENTI DI INFORMATICA 1
CORSI DI LAUREA IN INGEGNERIA CHIMICA E INGEGNERIA MECCANICA
6 giugno 2018

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

ESERCIZIO 1 (4 punti)

Descrivere in modo chiaro e sintetico l'architettura di un moderno sistema operativo.

ESERCIZIO 2 (4 punti)

Siano date le istanze di relazione:

Docenti

Nome_D	Cognome_D	Matricola	Corsi_tenuti
Gian Luca	Marcialis	1	2
Guido	Piano	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

Scrivere in SQL l'interrogazione: "Si trovi il nome e cognome dei docenti i cui studenti hanno registrato esami con voto superiore a 25".

ESERCIZIO 3 (25 punti)

(3 punti) Siete entrati in possesso dell'agenda del vostro migliore amico. Essa è conservata in un file "agenda.txt" nella quale ogni riga ha due elementi: il primo è la data di un certo evento, nel formato "gg.mm.anno" con due cifre per gg e mm, quattro cifre per anno (es. "12.03.2013"), e la seconda, separata dal carattere "-", è la descrizione dell'evento. Sfortunatamente, gli eventi sono inseriti in modo totalmente casuale, non ordinato, quindi reperire un evento particolare o capire quale sia il primo evento inserito è molto difficile. Scrivete dunque un programma Python che legge da file l'agenda e la memorizza in un dizionario, utilizzando come chiave le date rilevate e come valore una lista contenente i vari eventi. A questo punto interrogate il vostro dizionario inserendo da tastiera una data in modo tale il sistema scriva in un file "output.txt" gli eventi corrispondenti a quella data, se presenti, facendo seguire la stampa della prima data in ordine di tempo.

Esempio di file "agenda.txt":

```
20.12.2015-Dato esame di Fondamenti di Informatica
17.02.2014-Uscito con Marco e Fabiola
03.01.2018-Fidanzato con Sempronia
04.11.2017-Cenato da "McTizio"
03.01.2018-Dato esame di Analisi I
```

Corrispondente dizionario:

```
a={"20.12.2015": ["Dato esame di Fondamenti di Informatica\n"],
  "17.02.2014": ["Uscito con Marco e Fabiola\n"], "03.01.2018":
  ["Fidanzato con Sempronia\n", "Dato esame di Analisi I\n"],
  "04.11.2017": ["Cenato da 'McTizio'\n"]}
```

Se per esempio inseriamo da tastiera la data "05.11.2017", il file "output.txt" conterrà:

```
Data non presente
Prima data: 17.02.2014
```

Se invece si inserisce la data "03.01.2018":

```
Fidanzato con Sempronia
Dato esame di Analisi I
Prima data: 17.02.2014
```

Nell'implementare questo codice, si scrivano anche le seguenti funzioni:

(6 punti) Funzione `leggiAgenda(nomefile)`:

- Ingresso: il nome del file espresso come stringa
- Output: il dizionario che rappresenta l'agenda secondo il formato dato

Nota: ricordarsi che ogni fine riga include il carattere "\n" (v. esempio).

(5 punti) Funzione `confrontaDate(data1, data2)`:

- Ingresso: due stringhe rappresentante delle date nel formato "gg.mm.anno"
- Output: -1 se la prima stringa precede la seconda, 0 se sono identiche, 1 se la prima stringa segue la seconda.

(6 punti) Funzione `primaData(agenda)`:

- Ingresso: l'agenda espressa come dizionario
- Output: la prima data inserita in ordine di tempo nel formato "gg.mm.anno"

(5 punti) Funzione `scriviRisultato(agenda, data)`:

- Ingresso: l'agenda espressa come dizionario, una possibile data nel formato "gg.mm.anno"
- Output: non necessario

Scrive nel file "output.txt" l'evento o gli eventi corrispondenti alla data fornita, se presenti, seguiti dalla prima data utile presente nel dizionario.

Soluzione domanda 1.

V. dispense del corso.

Soluzione domanda 2.

“Si trovi il nome e cognome dei docenti i cui studenti hanno registrato esami con voto superiore a 25”

```
SELECT NomeD, CognomeD
FROM Allievi, Docenti
WHERE
    Docenti.Matricola=Allievi.Docente
AND
    MAX_VOTO>25
```

Soluzione domanda 3.

```
def leggiAgenda(nomefile):
    f=open(nomefile,"r")
    d={}
    r=f.readline()
    while r!="":
        data,evento=r.split("-")
        if data in d:
            d[data]=d[data]+[evento]
        else:
            d[data]=[evento]
        r=f.readline()
    f.close()
    return d

def confrontaDate(data1,data2):
    g1,m1,a1=data1.split(".")
    g2,m2,a2=data2.split(".")
    if int(a1)<int(a2):
        return -1
    elif int(a1)>int(a2):
        return 1
    elif int(m1)<int(m2):
        return -1
    elif int(m1)>int(m2):
        return 1
    elif int(g1)<int(g2):
        return -1
    elif int(g1)>int(g2):
        return 1

    return 0
```

```

def primaData(d):
    date=d.keys()
    pD=date[0]
    for data in date[1:]:
        if confrontaDate(data,pD)==-1:
            pD=data
    return pD

def scriviRisultato(d,data):
    pD=primaData(d)
    if data in d:
        s="".join(d[data])
    else:
        s="Data non presente\n"

    f=open("output.txt","w")
    f.write(s)
    f.write("Prima data: "+pD+"\n")
    f.close()

#Programma principale
ag=leggiAgenda("agenda.txt")
data=raw_input("Immetti una data:\n")
scriviRisultato(ag,data)

```