

**PROVA SCRITTA DEL CORSO DI**  
**FONDAMENTI DI INFORMATICA 1**  
**CORSO DI LAUREA IN INGEGNERIA CHIMICA ED INGEGNERIA MECCANICA**  
**3/2/2017**

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

**ESERCIZIO 1 (5 punti)**

Si consideri un valore a 4 bit pari 1100. Indicare il suo corrispondente valore intero: senza segno, in complemento a due, in eccesso 8.

**ESERCIZIO 2 (5 punti)**

Descrivere in modo chiaro e sintetico la CPU di un calcolatore elettronico.

**ESERCIZIO 3 (5 punti)**

Sia dato il seguente schema logico per due relazioni:

Relation Dischi (CodDisco: char(20), primary key not null, Titolo: char(50), CodAutore: integer, Anno: integer, CasaDiscografica: char(30))	Relation Autori (CodAutore: integer, primary key not null, Nome: char(50))
--	--

- 1) (2 punti) Indicare grado e cardinalità delle due relazioni
- 2) (3 punti) Esprimere in SQL l'interrogazione: "Indicare il titolo dei dischi registrati dai Pearl Jam per la Decca dal 1990."

#### ESERCIZIO 4 (18 punti)

(4 punti) Scrivere un programma in linguaggio Python che legga da un file "input.txt" una sequenza di linee ciascuna delle quali così organizzata:

<operazione operando1 operando2>

Dove:

"operazione" è un carattere pari a "+", "-", "\*", "/"

"operando1" e "operando2" sono i due termini dell'operazione: i due addenti, sottrattore e sottraendo, moltiplicando e moltiplicatore, dividendo e divisore. Possono essere una coppia di interi o una coppia di float, il risultato deve tenerne conto. Nel caso si tenti di dividere un valore per 0, nella riga corrispondente verrà scritto "Divisione per zero."

Per ogni riga del file, scriva su file output.txt il risultato.

Esempio:

Input.txt
+ 3 2
/ 3.0 2.0
/ 2 0

Output.txt
5
1.5
Divisione per zero

Si implementino inoltre le funzioni:

(4 punti) Funzione `leggi_operazioni` con:

- Parametro in ingresso: stringa indicante il `nome_file` da aprire;
- Parametri in uscita: una lista di liste, ognuna delle quali contenente l'operatore e i due operandi. Es. `[['+', 3, 2], ['/', 3.0, 2.0], ['/', 2, 0]]`.

La funzione legge dal file chiamato `nome_file` una sequenza di operazioni come descritto nell'esempio e le restituisce nella forma lista di liste.

(4 punti) Funzione `tipo` con:

- parametro in ingresso: una stringa che rappresenta un valore numerico
- parametro in uscita: il valore numerico in formato intero o in formato float a seconda del tipo. Si ricordi che un numero in formato float presenta sempre il separatore dei decimali seguito dalla parte frazionaria.

(3 punti) Funzione `esegui` con:

- parametro in ingresso: una lista contenente la sequenza operatore, operando1, operando2
- parametro in uscita: risultato dell'operazione

Riceve in ingresso una lista nella forma `[operatore, operando1, operando2]` ed esegue l'operazione indicata da `operatore`.

(3 punti) Funzione `scrivi_risultati`

- parametro in ingresso: una lista con i risultati delle singole operazioni
- parametro in uscita: nessuno.

Stampa su file "output.txt" il risultato di ciascuna operazione letta come da esempio.

**Soluzione dell'esercizio 1**

Configurazione binaria: 1100

Numeri naturali (interi senza segno)  $= 2^3 + 2^2 = 8 + 4 = 12$ .

Numeri in complemento a 2  $= -2^3 + 2^2 = -8 + 4 = -4$ .

Numeri in eccesso 8: valore naturale – eccesso (pari a  $1000_2 = 8_{10}$ )  $= 12 - 8 = 4$ .

**Soluzione dell'esercizio 2.** Vedi dispense, Cap. 2.

**Soluzione dell'esercizio n.3 (6 crediti)**

1. Il grado delle relazioni, dato dal numero di attributi, è rispettivamente 5 e 2. Non è possibile fornire la cardinalità in quanto essa si riferisce ad istanze di relazione, non a schemi.

2.

```
SELECT TITOLO
FROM DISCHI, AUTORI
WHERE DISCHI.CODAUTORE = AUTORI.CODAUTORE
AND NOME = 'PEARL JAM'
AND CASADISCOGRAFICA = 'DECCA'
AND ANNO >= 1990
```

#### Soluzione dell'esercizio 4.

Implementazione del programma:

#Programma per l'esecuzione di operazioni in sequenza

```
def main():  
    operazioni=leggi_operazioni("input.txt")
```

```
    risultati=[]  
    for operazione in operazioni:  
        risultato=esegui(operazione)  
        risultati=risultati+[risultato]
```

```
    scrivi_risultati(risultati)  
    return
```

```
def tipo(valore):
```

```
    val=valore.split('.')  
    if len(val)>1:  
        return float(valore)  
    else:  
        return int(valore)
```

```
def leggi_operazioni(nomefile):
```

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

```
    lista=[]  
    linea=idf.readline()  
    while (linea!=""):  
        linea=linea.split()  
        linea[1]=tipo(linea[1])  
        linea[2]=tipo(linea[2])  
        lista=lista+[linea]  
        linea=idf.readline()
```

```
    idf.close()  
    return lista
```

```
def esegui(operazione):
```

```
    operatore=operazione[0]  
    if operatore=='':  
        return operazione[1]*operazione[2]  
    elif operatore=='-':  
        return operazione[1]-operazione[2]  
    elif operatore=='*':  
        return operazione[1]*operazione[2]  
    elif operazione[2]==0:  
        return "Divisione per zero"  
    else:  
        return operazione[1]/operazione[2]
```

```
def scrivi_risultati(risultati):
```

```
    idf=open("output.txt","w")  
    for risultato in risultati:  
        idf.write(str(risultato)+"\n")  
    idf.close()
```