

**PROVA SCRITTA DEL CORSO DI  
FONDAMENTI DI INFORMATICA  
CORSI DI LAUREA IN INGEGNERIA CHIMICA ED INGEGNERIA MECCANICA  
22 Luglio 2015**

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

**ESERCIZIO 1 (5 punti)**

Si consideri il valore 111001.11 espresso in base 2.

- (1) Convertire il valore dato in base 10. (2 punti)
- (2) Estrarre la parte intera e calcolare la sua opposta utilizzando la rappresentazione in complemento a due con otto bit. (3 punti)

**ESERCIZIO 2 (5 punti)**

Spiegare in modo chiaro e sintetico qual è il compito del gestore dei processi in un moderno sistema operativo.

**ESERCIZIO 3 (5 punti)**

<p>Relation Studenti (Matricola: char(5) PRIMARY KEY NOT NULL, Nome: char(20) NOT NULL, Data-N: date, Anno-Corso: integer, Corso-Laurea: char (1));</p> <p>Relation Corsi (Cod-Corso: char(6) PRIMARY KEY NOT NULL, Titolo: char(50), Docente: char(20));</p> <p>Relation Esami (Cod-Corso: char(5) NOT NULL, Matricola: char(5) NOT NULL, Data-E: date NOT NULL, Voto: integer NOT NULL, PRIMARY KEY (Cod-Corso, Matricola));</p>	<p>Facendo riferimento allo schema logico a sinistra:</p> <p>(2 punti) Aggiungere alla tabella Corsi la seguente tupla: (1, Fondamenti di Informatica, Paperino).</p> <p>(3 punti) Scrivere in SQL la seguente interrogazione, precisandone il tipo: Estrarre nome e matricola degli studenti che hanno sostenuto il corso con codice pari a 1 prendendo un voto superiore a 25.</p>
--	--

**ESERCIZIO 4 (18 punti)**

1. (2 punti) Scrivere uno script Python che, ricevendo da tastiera un valore intero ed una stringa, li memorizzi in una stringa.
2. (4 punti) Scrivere un frammento di codice Python che, a partire da una lista di stringhe, stampi a video la stringa più lunga. (Suggerimento: per la soluzione si usi la funzione len(s) che applicata alla stringa s restituisce il numero di caratteri nella stringa).
3. (5 punti) Scrivere una funzione python esegui\_operazione che ricevendo in ingresso un carattere pari ad uno dei simboli '+', '-', '\*', '/' e due valori numerici, restituisca l'operazione corrispondente al simbolo, considerando il primo ed il secondo valore come primo e secondo operando rispettivamente (ad esempio, nel caso del simbolo '/', il primo valore è il dividendo ed il secondo è il divisore).
4. (7 punti) Scrivere uno script Python che, a partire dal file "dati.txt", legga una lista di triple formate nell'ordine da uno tra i simboli '+', '-', '\*', '/', e due valori numerici. Il file ad esempio è formattato come segue:  
+ 3.1 5.0  
/ 5.0 2.5

Per ognuna di queste righe, il programma deve salvare su file "output.txt" il risultato delle operazioni rappresentate dai simboli, considerando il primo e secondo valore come relativi operandi. Si usi la funzione sviluppata nell'esercizio precedente. Nel caso dell'esempio, il file output conterrà i valori salvati come segue:

8.1  
2.0

## Soluzioni

### ESERCIZIO 1 (5 punti)

Si consideri il valore 111001.11 espresso in base 2.

- (1) Convertire il valore dato in base 10. (2 punti)
- (2) Estrarre la parte intera e calcolare la sua opposta utilizzando la rappresentazione in complemento a due con otto bit. (3 punti)

(1) Formula di conversione binario-decimale:

$$v = \sum_{i=-M}^{N-1} b_i 2^i$$

Con N numero di bit in parte intera, M numero di bit in parte frazionaria,  $b_i$  il bit nella posizione  $i$  considerando  $i=0$  il primo bit a sinistra dopo il separatore decimale.

Applicando questa formula al caso in oggetto, dove  $N=5$  e  $M=2$ , si ottiene:  $111001.11_2 = 57.75_{10}$ .

- (2) Innanzi tutto estraiamo la parte intera e rappresentiamola con otto bit: 00111001. Per calcolare il complemento a due invertiamo prima di tutto i singoli bit ottenendo 11000110 ed aggiungiamo 1, ottenendo 11000111. Per verificare che il calcolo sia corretto è sufficiente sommare il valore ottenuto a quello dato, ottenendo la somma a otto bit pari a un ottetto di zeri.

### ESERCIZIO 2 (5 punti)

Per la definizione gestore dei processi e suoi compiti, vedere dispense del corso.

### ESERCIZIO 3 (5 punti)

(1)

```
INSERT INTO Corsi VALUES ((1, Fondamenti di Informatica, Paperino))
COMMIT WORK
```

(2)

```
SELECT NOME, MATRICOLA
FROM STUDENTI, ESAMI
WHERE STUDENTI.MATRICOLA=ESAMI.MATRICOLA
AND COD-CORSO=1
AND VOTO>25
```

Si tratta senza dubbio di una join, perché richiede un sottoinsieme del prodotto cartesiano delle relazioni Studenti ed Esami.

#### ESERCIZIO 4 (18 punti)

1. (2 punti) Scrivere uno script Python che, ricevendo da tastiera un valore intero ed una stringa, li memorizzi in una lista.

```
valore_intero=raw_input()
stringa=raw_input()
l=[int(valore_intero), stringa]
```

2. (4 punti) Scrivere un frammento di codice Python che, a partire da una lista di stringhe, stampi a video la stringa più lunga. (Suggerimento: per la soluzione si usi la funzione `len(s)` che applicata alla stringa `s` restituisce il numero di caratteri nella stringa).

```
max_lunghezza=0
max_stringa=""
for stringa in lista_stringhe:
    lunghezza_stringa=len(stringa)
    if lunghezza_stringa>max_lunghezza:
        max_lunghezza=lunghezza_stringa
        max_stringa=stringa
print max_stringa
```

3. (5 punti) Scrivere una funzione python `esegui_operazione` che ricevendo in ingresso un carattere pari ad uno dei simboli '+', '-', '\*', '/' e due valori numerici, restituisca l'operazione corrispondente al simbolo, considerando il primo ed il secondo valore come primo e secondo operando rispettivamente (ad esempio, nel caso del simbolo '/', il primo valore è il dividendo ed il secondo è il divisore).

```
def esegui_operazione operatore, operandol, operando2):
    if operatore=="+":
        return operandol+operando2
    if operatore=="-":
        return operandol-operando2
    if operatore=="*":
        return operandol*operando2
    if operatore=="/":
        return operandol/operando2
```

4. (7 punti) Scrivere uno script Python che, a partire dal file "dati.txt", legga una lista di triple formate nell'ordine da uno tra i simboli '+', '-', '\*', '/', e due valori numerici. Il file ad esempio è formattato come segue:

```
+ 3.1 5.0  
/ 5.0 2.5
```

Per ognuna di queste righe, il programma deve salvare su file "output.txt" il risultato delle operazioni rappresentate dai simboli, considerando il primo e secondo valore come relativi operandi. Si usi la funzione sviluppata nell'esercizio precedente. Nel caso dell'esempio, il file output conterrà i valori salvati come segue:

```
8.1  
2.0
```

```
infile=open("dati.txt","r")  
outfile=open("output.txt","w")  
linea=infile.readline()  
while linea!="":  
    argomenti=linea.split()  
    operatore=argomenti[0]  
    operando1=float(argomenti[1])  
    operando2=float(argomenti[2])  
    risultato=esegui_operazione(operatore,operando1,operando2)  
    outfile.write(str(risultato)+"\n")  
    linea=infile.readline()  
infile.close()  
outfile.close()
```