

SIMULAZIONE SECONDA PROVA INTERMEDIA DEL CORSO DI
CORSI DI LAUREA IN INGEGNERIA CHIMICA ED INGEGNERIA MECCANICA
25/05/2016

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

NOME: _____ **COGNOME:** _____ **MATRICOLA:** _____

ESERCIZIO 1: RETI DI CALCOLATORI (5 punti)

Descrivere il protocollo TCP/IP.

ESERCIZIO 2: ALGORITMI E LINGUAGGI (5 punti)

Definire le proprietà degli algoritmi. Utilizzare anche qualche esempio.

ESERCIZIO 3: PYTHON (23 punti)

(1) (3 punti) Scrivere un programma Python che data la stringa 'Mario Rossi 2016' stampi la scritta 'Rossi Mario 2016'. (Nota: nome, cognome e anno sono separati da uno spazio)

(2) Data una lista contenente i prodotti presenti in un magazzino e una lista contenente le corrispettive quantità depositate in magazzino:

(esempio lista_p=['frigorifero', 'lavatrice', 'aspirapolvere'] e lista_q=[20,32,15]):

- i. (4 punti) Scrivere una funzione che preso in ingresso la lista contenente la quantità dei prodotti, restituisca la quantità totale di merce presente in magazzino.
- ii. (4 punti) Scrivere una funzione che preso in ingresso entrambe le liste mi restituisca il prodotto più venduto (ossia quello meno presente in magazzino). Si supponga che non esistano prodotti in giacenza in medesima quantità (non si verificherà ad esempio che lista_p=['frigorifero' , 'lavatrice' , 'aspirapolvere'] e lista_q=[20,20,15], ovvero che il numero di prodotti in giacenza di 'frigorifero' e 'lavatrice' siano entrambi eguali, come nell'esempio, a 20).
- iii. (4 punti) Scrivere una funzione che preso come argomento le due liste e un prodotto, restituisca quanti prodotti sono presenti in magazzino. Se il prodotto non è presente in magazzino restituisca 0.
- iv. (3 punti) Scrivere una funzione Python che legga da file fornito come parametro della funzione, una sequenza di oggetti scritti, linea per linea nel modo esemplificato:

```
frigorifero 20
lavatrice 32
aspirapolvere 15
...
```

e restituisca in uscita due liste, una relativa alla sequenza dei prodotti, l'altra relativa alla sequenza delle giacenze.

- v. (5 punti) Utilizzando le funzioni implementate nei quattro punti precedenti, scrivere un programma che permetta di verificare di quanti pezzi del prodotto meno presente in magazzino (lista di prodotti e quantità tutte in un file 'magazzino.txt') è possibile rifornire il magazzino considerando che la capienza massima del magazzino è di 100 pezzi.

Soluzione dell'esercizio 1 Vedi dispense, Cap. 5, pagg. 31-35

Soluzione dell'esercizio 2. Vedi dispense, Cap. 6, pagg.5-8.

Soluzione dell'esercizio 3

Risposta 3.1 souzione1

```
dato='Mario Rossi 2016'
lista_dato=dato.split(' ')
temp=lista_dato[0]
lista_dato[0] = lista_dato[1]
lista_dato[1] = temp
dato_new=' '.join(lista_dato)
print dato_new
```

Risposta 3.1 souzione2

```
dato='Mario Rossi 2016'
lista_dato=dato.split(' ')
print lista_dato[1]+" "+ lista_dato[0]+" "+ lista_dato[2]
```

Risposta 3.2

i.

```
def calcola_giacenza(lista_q):
    conta=0
    for elem in lista_q:
        conta=conta+elem
    return conta
```

ii. soluzione 1

```
def trova_piu_venduto(lista_p,lista_q):
    min_lista_p=lista_p[0]
    min_lista_q=lista_q[0]
    i=1
    while i<len(lista_p):
        if lista_q[i]<min_lista_q:
            min_lista_q=lista_q[i]
            min_lista_p=lista_p[i]
        i=i+1
    return min_lista_p
```

ii. soluzione 2

```
def trova_piu_venduto(lista_p,lista_q):
    pos_min=0
    min_lista_q=lista_q[0]
    i=1
    while i<len(lista_p):
        if lista_q[i]<min_lista_q:
            min_lista_q=lista_q[i]
            pos_min=i
        i=i+1
    return lista_p[pos_min]
```

iii. soluzione 1

```
def presenza_magazzino(lista_p,lista_q,oggetto):
    i=0
    if oggetto not in lista_p:
        return 0
    else:
        while i<len(lista_p):
            if oggetto==lista_p[i]:
                return lista_q[i]
            i=i+1
```

iii. soluzione 2

```
def presenza_magazzino(lista_p, lista_q, oggetto):  
    d={}  
    i=0  
    for p in lista_p:  
        d[p]=lista_p[i]  
        i=i+1  
    if oggetto in d:  
        return d[prodotto]  
    else:  
        return 0
```

iii. soluzione 3

```
def presenza_magazzino(lista_p, lista_q, oggetto):  
    n=len(lista_p)  
    i=0  
    while (lista_p[i]!=oggetto):  
        i=i+1  
    if i==n:  
        return 0  
    else:  
        return lista_q[i]
```

iii. soluzione 4

```
def presenza_magazzino(lista_p, lista_q, oggetto):  
    if oggetto not in lista_p:  
        return 0  
    else:  
        i=0  
        n=len(lista_p)  
        trovato=False  
        while i<n and (not trovato):  
            trovato= lista_p[i]==oggetto  
            i=i+1  
        return lista_q[i]
```

iv.

```
def leggi_file(nomefile):  
    mio_file=open(nomefile,"r")  
    linea=mio_file.readline()  
    lista_prodotti=[]  
    lista_quant=[]  
    while linea!='':  
        lista_linea=linea.split(' ')  
        lista_prodotti = lista_prodotti+ [lista_linea[0]]  
        lista_quant = lista_quant+ [int(lista_linea[1])]  
        linea=mio_file.readline()  
  
    mio_file.close()  
    return lista_prodotti, lista_quant
```

v.

```
max_c=100  
lista_p , lista_q = leggi_file('magazzino.txt')  
totale_pezzi_magazzino = calcola_giacenza (lista_q)  
if totale_pezzi_magazzino == max_c:  
    print 'Non puoi rifornire il magazzino con altri pezzi'  
else:  
    prodotto_verifica_acquisto= trova_piu_venduto(lista_p, lista_q)  
    quant = presenza_magazzino(lista_p, lista_q, prodotto_verifica_acquisto)  
    print 'Puoi acquistare altri', max_c-totale_pezzi_magazzino , 'pezzi di',  
    prodotto_verifica_acquisto
```