

# Esercizio visto a lezione

```
def max_o_min(lista,ord):
    lunghezza = len(lista)
    if (lunghezza > 0):
        m = lista[0]
        posizione = 1
        while (posizione < lunghezza):
            if (ord == 0) : []
                # se vero m = minimo
                if (lista[posizione] < m):
                    m = lista[posizione]
            else:
                # altrimenti m = massimo
                if (lista[posizione] > m):
                    m = lista[posizione]
            posizione = posizione + 1
    return m
```

```
def ordina(lista,ordine):
    nuova_lista = [ ]
    while (len(lista) > 0):
        if (ordine == 'd'):
            m = max_o_min(lista,1)
        else:
            m = max_o_min(lista,0)
        nuova_lista.append(m)
        lista.remove(m)
    return nuova_lista
```

# Leggere e scrivere file

- Guardiamo il libro online: Capitolo 11, pagina 124 del file pdf

## Esercizio di programmazione

**Scrivere un programma che legge una lista numerica da un file, il cui nome è passato come primo parametro, e scrive gli elementi della lista, dal valore più grande al valore più piccolo, in un altro file, il cui nome è passato come secondo parametro**

### Variante:

**passare un terzo parametro che stabilisca se ordinare gli elementi in modo crescente o decrescente**

# Esempio di soluzione

All'inizio del file utilizzare il comando

```
import sys
```

che permette di accedere ai parametri passati a python da linea di comando.

Quindi seguono le definizioni delle funzioni `max_o_min` e `ordina`.

Dopo tali definizioni va inserito il codice mostrato di seguito.

```
f_in = open(sys.argv[1], "r")
lista_in = f_in.readlines()
f_in.close()
lista_num = [ ]
for elemento in lista_in :
    lista_num.append(int(elemento))
lista_ord = ordina(lista_num, sys.argv[3])
f_out = open(sys.argv[2], "w")
for num in lista_ord :
    f_out.write(str(num) + "\n")
f_out.close()
```

# Altre strutture dati

- **Tuple**

- Guardiamo il libro online: Capitolo 9, pagina 108 del file pdf

- **Dizionari**

- Guardiamo il libro online: Capitolo 10, pagina 116 del file pdf

# Esercizio

**Scrivere una funzione `istogramma` che data una stringa in input restituisca una lista di caratteri che compaiono nella stringa con la frequenza associata.**

Aiuto: utilizzare un dizionario.

```
def istogramma(stringa_in_input):
```

```
    corpo della funzione
```

```
    return lista_istogramma
```

Esempio di chiamata

```
str = "esempio di stringa"
```

```
L = istogramma(str) # oppure: istogramma("esempio di stringa")
```

```
L avrà valore ['e',2,'s',2,'m',1,'p',1,'i',3,'o',1,' ',2,'t',1,'r',1,'n',1,'g',1,'a',1]
```

# Possibile soluzione

funzione che restituisce una lista che alterna caratteri a frequenze;  
funzione che restituisce una lista di coppie (carattere, frequenza)

```
def istogramma(stringa_in_input):
    ist = {}
    for carattere in stringa_in_input :
        if ist.has_key(carattere) :
            ist[carattere] = ist[carattere] + 1
        else :
            ist[carattere] = 1
    lista_istogramma = []
    for key in ist.keys() :
        lista_istogramma.append(key)
        lista_istogramma.append(ist[key])
    return lista_istogramma

def istogramma_coppie(stringa_in_input): # restituisce lista di coppie
    ist = {}
    for carattere in stringa_in_input :
        if ist.has_key(carattere) :
            ist[carattere] = ist[carattere] + 1
        else :
            ist[carattere] = 1
    lista_istogramma = []
    for key in ist.keys() :
        lista_istogramma.append((key,ist[key]))
    return lista_istogramma
```