

Menù

- **Esercizi sulle liste**
- **Esercizi sulle liste annidate**

Esercizi sulle liste

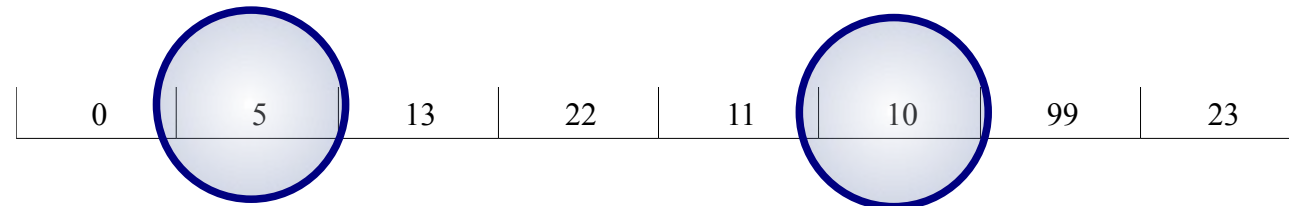
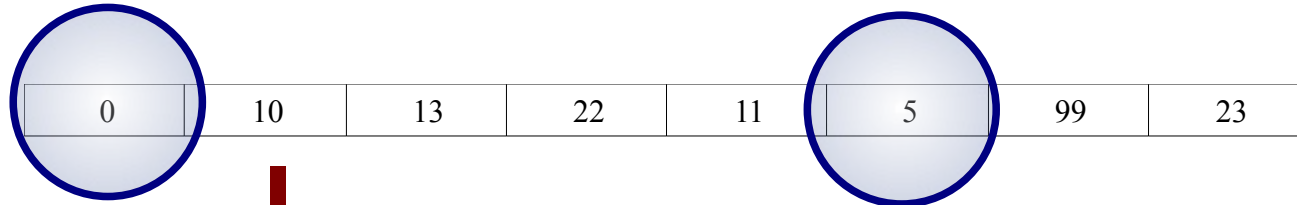
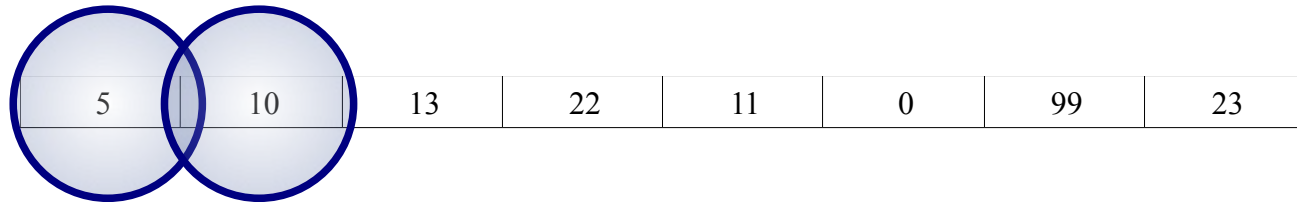
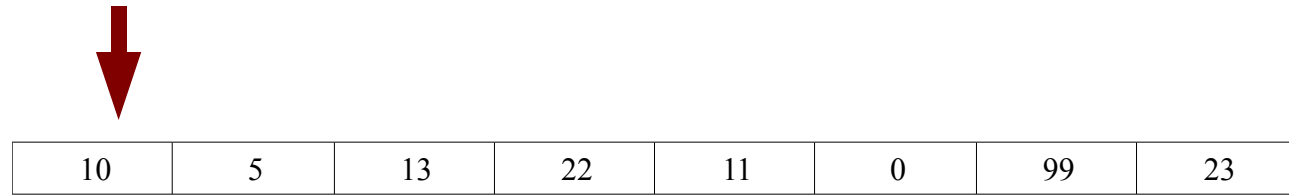
1 di 3

Realizza una funzione che prenda in input una lista di numeri interi L e li ordini.

Metodo inefficiente ma **intuitivo**

```
def ordina (L):  
    N = len(L)  
    if (N <= 1):  
        return L  
    i = 0  
    while (i < N):  
        j = i+1  
        while (j <= N-1):  
            if (L[i] > L[j]):  
                X = L[i]  
                L[i] = L[j]  
                L[j] = X  
            j = j+1  
        i = i+1  
    return L
```

Esercizi sulle liste



.....

Esercizi sulle liste

3 di 3

Generalizza la funzione precedente in modo che l'ordinamento possa essere sia crescente sia decrescente in base al valore associato a un secondo parametro

```
def ordina (L,ORDI):
    N = len(L)
    if (N <= 1):
        return L
    i = 0
    while (i < N-1):
        j = i+1
        while (j <= N-1):
            if (L[i]> L[j] and ORDI==1) or (L[i]< L[j] and ORDI==0):
                X = L[i]
                L[i] = L[j]
                L[j] = X
            j = j+1
        i = i+1
    return L
```

Esercizi sulle liste annidate

1 di 2

Realizza una funzione per il riempimento automatico di una matrice $N \times M$ (realizzata come lista annidata)

```
import random
```

```
def crea (N,M):  
    c=[0]*N  
    i = 0  
    while (i<N):  
        c[i]=[0]*M  
        i=i+1  
    return c
```

```
def randmatrix(N,M,A,B):  
    MTX=crea(N,M)  
    i=0  
    while (i<N):  
        j=0  
        while (j<M):  
            MTX[i][j]=int(A+B*random.random())  
            j=j+1  
        i=i+1  
    return MTX
```

Esercizi sulle liste

2 di 2

Realizza una funzione per la somma tra due matrice $N \times M$ (realizzata come lista annidata):

```
def sum_matrix(A,B):
    N1=len(A)
    N2=len(B)
    M1=len(A[0])
    M2=len(B[0])
    if N1 != N2:
        return 0
    if M1 != M2:
        return 0
    MTX=crea(N1,M1)
    i=0
    while (i<N1):
        j=0
        while (j<M1):
            MTX[i][j]=A[i][j]+B[i][j]
            j=j+1
        i=i+1
    return MTX
```