

# Liste

Es 1:

Scrivere un programma in grado di svuotare due liste singolarmente concatenate di interi (L1, L2), ordinate in modo decrescente, e di fonderle in una singola lista ordinata di interi L3 ordinata in modo decrescente. La funzione `mergelists` deve ricevere come parametri i riferimenti alle liste L1, L2, L3.

Es 2:

Scrivere una funzione membro della classe `Alist` per invertire una lista.

Es 3:

Leggere da un file una sequenza di N numeri floating point (il file contiene un numero per ogni riga, la prima riga del file contiene il valore N) e rappresentarla con una lista L; stampare la lista.  
Costruire una seconda lista L2 composta dai soli elementi di L maggiori della media; stampare L2.

Es 4:

Creare una lista di numeri interi ordinati in ordine crescente. Dopo aver memorizzato la sequenza in una lista, inserire nella posizione corretta all'interno della lista, tutti i numeri mancanti.  
Stampare in output la lista. Non devono essere usate altre liste o array di appoggio.

Esempio: Supponiamo che sia fornita in input la sequenza 4, 7, 8, 9,15,17,21. Dopo aver memorizzato gli elementi nella lista 4 -> 7 -> 8 -> . . . > 21

vengono inseriti i numeri mancanti, ottenendo la lista composta dagli elementi

4 -> 5 -> 6 -> 7 -> 8 -> . . . -> 19 -> 20 -> 21

Es 5:

Scrivere una funzione che elimini tutti i numeri duplicati da una lista singolarmente concatenata contenente numeri interi. La lista non è ordinata.

Es 6:

Scrivere un algoritmo per invertire una lista singolarmente concatenata che utilizzi i metodi `moveToPos` e `remove`.