

14.14 Esercizi.

- * 1. Scrivere un programma che accetti in ingresso una sequenza di valori interi terminante con zero e la memorizzi in una lista lineare. Successivamente il programma deve determinare il numero di pari e di dispari presenti nella lista. Risolvere il problema anche con funzioni ricorsive.
- * 2. Scrivere un programma che accetti in ingresso una sequenza di valori interi terminante con zero e la memorizzi in una lista lineare. Successivamente il programma deve eliminare dalla lista i numeri pari.
- 3. Scrivere un programma che accetti in ingresso una sequenza di valori interi terminante con zero e la memorizzi in una lista lineare. Successivamente eliminare dalla lista creata quelli che non sono divisori di n , dove n è un numero intero passato in ingresso dall'utente.
- * 4. Data in ingresso una sequenza di valori interi terminante con zero, costruire due liste lineari, una contenente i valori positivi e una i valori negativi. Visualizzare le liste costruite.
- 5. Definire una funzione che calcoli il numero totale degli elementi che compongono una lista. Scrivere un'altra funzione che stampi l' n -esimo elemento di una lista se questo esiste, altrimenti visualizzi il messaggio: "La lista non contiene un numero sufficiente di elementi".
- 6. Scrivere una funzione che cancelli da una lista tutte le occorrenze di un particolare elemento se questo è presente nella lista, altrimenti visualizzi il messaggio: "Elemento non presente nella lista".
- 7. Realizzare una funzione che ordini una lista in modo crescente. Scrivere un'altra funzione che inserisca in una lista ordinata al posto opportuno un nuovo elemento richiesto all'utente.
- 8. Scrivere una funzione che visualizzi in ordine inverso una lista.
- 9. Scrivere una funzione che inverta l'ordine di una lista.
- 10. Scrivere una funzione che, a partire da due liste, ne costruisca una terza ottenuta alternando gli elementi delle altre due.
- 11. Scrivere la dichiarazione di una lista bidirezionale, ovvero di una lista in cui ogni elemento, oltre al campo informazione e a quello puntatore all'elemento successivo, ha anche un campo puntatore all'elemento precedente. Definire quindi una struttura composta da elementi di questo tipo cui si può accedere grazie a due puntatori: uno al primo elemento e uno all'ultimo.
- 12. Scrivere un programma completo per la gestione tramite vettore di una coda. In particolare, il programma dovrà gestire le operazioni di inserimento, cancellazione e visualizzazione della coda.
- 13. Realizzare la gestione di una coda che comprenda le operazioni di inserimento, cancellazione e visualizzazione, implementando la struttura dati mediante una array circolare.
- 14. Ripetere l'esercizio precedente, implementando la struttura dati mediante una lista circolare.
- 15. Aggiungere alle soluzioni dei due esercizi precedenti le funzioni che permettono di verificare se un certo valore passato in ingresso dall'utente è presente nella coda.