

## Esercitazione 7

Prof. Riccardo Zese - [riccardo.zese@unife.it](mailto:riccardo.zese@unife.it)

Ing. Alice Bizzarri - [alice.bizzarri@unife.it](mailto:alice.bizzarri@unife.it)

Ing. Elisabetta Gentili - [elisabetta.gentili1@unife.it](mailto:elisabetta.gentili1@unife.it)

Laboratorio Fondamenti di Informatica - Modulo B

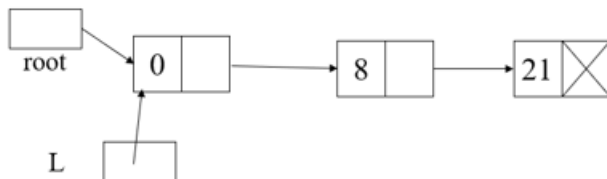


**DE** Department of  
Engineering  
Ferrara

## Creazione di una Lista l

Si sviluppi un programma che legga da tastiera una sequenza di numeri interi terminata da 0. Inserire poi ogni elemento letto in una lista utilizzando **l'inserimento in testa**. La funzione deve avere il seguente prototipo: `list cons(int n, list L)`.

Per ogni inserimento, **malloc** e aggiustamento dei puntatori.



root inizialmente **NULL**.

## Creazione di una Lista II

```
1  #include <stdlib.h>
2  #include <stdio.h>
3  typedef struct list_element {
4      int value;
5      struct list_element *next;
6  } item;
7  typedef item *list;
8
9  int main() {
10     list L = NULL; int i;
11     do {
12         printf("\n Introdurre valore (0 per
13             terminare): ");
14         scanf("%d", &i);
15         L = cons(i, L); // da implementare
16         // gestire l'inserimento del valore 0
17     } while (i!=0);
18 }
```

## Stampa di una Lista

Partendo dalla lista precedente, stamparne il contenuto utilizzando una funzione (procedura) con il seguente prototipo: void showlist(list l).

```
1 // se avessimo un vettore
2 i=0;
3 while (i<N) {
4     printf("\nValore : \t%d", V[i]);
5     i++;
6 }
```

Progettarne una versione **ITERATIVA** e una **RICORSIVA**.

## Ricerca Elemento

Definire una funzione member che, data una lista l ed un elemento n, restituisca l'indice dell'elemento (-1 se non trovato).

```
1 printf("\n Introdurre un valore da cercare: \t");
2 scanf("%d", &n);
3 i = member(n, l);
4 if (i >= 0)
5     printf("Trovato all'indice: %d\n", i);
6 else
7     printf("Non trovato\n");
```

Non abbiamo a disposizione un *indice* da incrementare per scandire la lista, come succedeva per la rappresentazione sequenziale mediante vettori.

La lista va scandita (sequenzialmente) con un **puntatore**.

Progettarne una versione **ITERATIVA** e una **RICORSIVA**.

## Esempio

Se la sequenza in ingresso è: 1,2,3,4,5,6,(0), il comportamento delle funzioni dovrà essere il seguente:

- `cons (+ showlist)`: 6 5 4 3 2 1
- `member(n,l)`:
  - inserisci un intero da cercare nella lista: 1
  - Trovato nella posizione 5
  - inserisci un intero da cercare nella lista: 7
  - Non trovato

Strutturare il progetto su più file.

# Definizione iterativa e ricorsiva I

In lista.h:

```
1 #define RECURSIVE
```

In lista.c

```
1 #ifndef RECURSIVE
2 // Tutte le funzioni NON
3 // ricorsive
4 #endif
5
6 #ifdef RECURSIVE
7 // Tutte le funzioni
8 // ricorsive
9 #endif
```

## Definizione iterativa e ricorsiva II

File lista.h:

```
1  #define RECURSIVE
2
3  typedef struct node{
4      int n;
5      struct node *next;
6  } listNode;
7  typedef listNode* list;
8  typedef enum {FALSE, TRUE} boolean;
9
10 list cons(int n, list l);
11 void showlist(list l); //R+IT
12 int member(int n, list l); //R+IT
```