

## 10. ESERCIZI funzioni e procedure su liste

Gli esercizi hanno lo scopo di esercitarsi nell'utilizzo dei costrutti del linguaggio visti a lezione, nella fattispecie: i tipi `int` e `float` con le operazioni aritmetiche, le dichiarazioni di variabili, l'assegnamento, le funzioni `printf` e `scanf` e le costanti, le istruzioni condizionali e gli iteratori, gli array a una o più dimensioni, i puntatori, le funzioni e le procedure e le liste allocate dinamicamente (Eccetto specifici casi in cui qualche costrutto è esplicitamente vietato). Si raccomanda allo studente di non usare costrutti non ancora spiegati perchè l'esercizio perderebbe il suo scopo pedagogico. È anche vietato usare funzioni di libreria non esplicitamente richiamate nel testo.

Nelle descrizioni delle funzioni, la specifica dei parametri non è esaustiva, ulteriori parametri possono essere aggiunti se necessario. Tutti gli esercizi che richiedono la scrittura di funzioni devono essere completati con un programma principale che permetta di verificare la correttezza della funzione. Per provare le funzioni si usi inizialmente la procedura `readListIntRic` che legge `K` interi dall'input e crea una lista contenente `K` elementi (i valori sono memorizzati nell'ordine inverso a quello in cui sono stati letti) di seguito definita.

```
void readListIntRic (ListaDiElementi *L, int K ) {
    if (K>0)
    { int x;
      printf("Digita un intero elemento della lista\n");
      scanf("%d",&x);
      *L=inTesta(*L,x);
      readListIntRic(L,K-1);
    } }
```

La funzione `inTesta` è definita di seguito

```
ListaDiElementi inTesta(ListaDiElementi L, int v){
    ListaDiElementi aux=malloc(sizeof(ElementoLista));
    aux->info=v;
    aux->next=L;
    return aux;
}
```

### ESERCIZIO 10.1

Si scriva in C una funzione ricorsiva con un parametro `L` lista di interi, che stampa in output gli elementi di `L` separati da uno spazio e chiusi tra parentesi.

### ESERCIZIO 10.2

Si scriva in C una funzione iterativa con un parametro `L` lista di interi, che stampa in output gli elementi di `L` separati da uno spazio e chiusi tra parentesi.

### ESERCIZIO 10.3

Si scriva in C una funzione ricorsiva con un parametro `L` lista di interi che calcola il valore minimo contenuto nella lista.

### ESERCIZIO 10.4

Si scriva in C una funzione iterativa con un parametro `L` lista di interi che calcola il valore minimo contenuto nella lista.

### ESERCIZIO 10.5

Si definisca in C un funzione ricorsiva con un parametro `L` lista di interi ed un parametro `x` di tipo `int` e un parametro `p` di tipo `*int` che calcola true se la lista contiene almeno un elemento uguale ad `x` e restituisce nel parametro `p` (passato per indirizzo) la posizione (della prima occorrenza) di `x` nella lista.

### ESERCIZIO 10.6

Si definisca in C una funzione iterativa con un parametro `L` lista di interi ed un parametro `x` di tipo `int` e un parametro `p` di tipo `*int` che calcola true se la lista contiene almeno un elemento uguale ad `x` e restituisce nel parametro `p` (passato per indirizzo) la posizione (della prima occorrenza) di `x` nella lista.

### ESERCIZIO 10.7

Si scriva in C la funzione iterativa della `readlistIntRic` definita sopra.

**ESERCIZIO 10.8**

Si scriva in C la versione che passa il parametro lista per valore e restituisce come risultato la lista.

**ESERCIZIO 10.9**

Si modifichi la funzione `readListIntRic` affinché utilizzi la procedura `IserisciTestalista` contenuta nei lucidi visti a lezione.

**ESERCIZIO 10.10**

Si scriva in C una funzione ricorsiva, con un parametro L lista di interi che copia gli elementi contenuti in L in una nuova lista il cui puntatore al primo elemento viene restituito come risultato della funzione.

**ESERCIZIO 10.11**

Si scriva in C una funzione iterativa, con un parametro L lista di interi che copia gli elementi contenuti in L in una nuova lista il cui puntatore al primo elemento viene restituito come risultato della funzione.