

# Esercizi di programmazione per il corso di Algoritmi

## Lezione 7-8

### Esercizi sulle liste

**Esercizio 1** Si scriva un codice `progress.c`, che legge da tastiera un numero  $n$ , inserisce in cima a una lista  $L$  i primi  $n$  numeri positivi e li stampa nell'ordine che seguono lungo la lista.

**Esercizio 2** Si aggiunga al codice precedente una funzione che scorre la lista  $L$  per cercare la posizione occupata dal numero  $n/2$  e poi stampa il contenuto della lista da quella posizione alla fine.

**Esercizio 3** Si implementi una libreria per gestire liste di numeri interi, costituita dai file `listaint.c` e `listaint.h`. Si usi tale libreria per leggere da file un insieme di interi aggiungendo i valori letti alla lista in cima, oppure in fondo, oppure in posizione crescente o decrescente.

Si aggiungano quindi alla libreria:

- una funzione di inserimento ordinato (nel caso la lista sia bidirezionale, si può anche scegliere se aggiungere l'elemento dalla cima o dal fondo per confronto col primo e con l'ultimo elemento della lista);
- una funzione di concatenamento di liste;
- una funzione di conteggio del numero di elementi della lista;
- una funzione di ricerca che restituisce la posizione di un elemento in una lista.

**Esercizio 4** Si scriva un programma `listamult.c` che gestisce su una tabella (quindi con un'implementazione a indici) un numero variabile di liste (al massimo 10) e obbedisca a comandi del tipo<sup>1</sup>:

- aggiungere un elemento a una lista di indice dato (da 1 a 10);
- stampare una lista di indice dato;
- contare gli elementi di una lista di indice dato;
- concatenare due liste di indice dato;
- cercare elementi in una lista;
- cancellare una lista;
- ordinare le liste per cardinalità decrescente;
- ecc. . .

---

<sup>1</sup>Questo esercizio è un'idea peregrina buttata lì per stimolare la fantasia.

**Esercizio 5** Dato il vettore `int V[8]` di valore `[ 5 15 34 54 14 2 52 72 ]` e i due puntatori `int *p, *q`, inizializzati rispettivamente con  
`p = &V[1];`  
`q = &V[5];`

- quanto vale `*(p+3)`?
- quanto vale `*(q-3)`?
- quanto vale `q-p`?
- è vero o falso che `p < q`?
- è vero o falso che `*p < *q`?

**Esercizio 6** Supponendo che `s`, `d` e `m` siano puntatori a elementi di un vettore, si vuole scrivere un'istruzione che faccia puntare `m` all'elemento intermedio fra `s` e `d` (nel caso vi siano due elementi intermedi, si consideri quello di sinistra; ad esempio, per `s = &V[3]` e `d = &V[6]` sia `m = &V[4]`). L'istruzione `m = (s + d)/2;` è scorretta. Perché? Come ottenere un'istruzione corretta, usando l'aritmetica dei puntatori?

**Esercizio 7** Sia `int V[10]; int *p; e p = V;`  
Indicare se le seguenti espressioni sono lecite o no, e per quelle lecite se sono vere o false:

- `p == V[0]`
- `p == &V[0]`
- `*p == V[0]`
- `p[0] == V[0]`