

## ESERCIZI - STRUTTURE

### 1. LISTA LINEARE (o lista linkata) (soluzione in *list.c*):

- (a) scrivere un programma che crei una lista lineare di elementi omogenei *floating point* che occupano in memoria una posizione qualsiasi (non contigua). Si ha dunque che la dichiarazione di ogni elemento è:

```
struct list_element {float value; struct list_element *pun};
```

Il campo del puntatore dell'ultimo elemento della lista non fa riferimento a nessun altro elemento, quindi gli viene assegnato il valore NULL;

- i. implementare una funzione che crei la lista con un numero di elementi  $N \geq 2$ ;
  - ii. implementare una funzione che inizializzi ciascun elemento della lista con un numero random nell'intervallo  $[0, 1)$ ;
  - iii. implementare una funzione che visualizzi a schermo i valori degli elementi della lista;
  - iv. implementare una funzione che determini il massimo ed il minimo della lista e li visualizzi;
  - v. con una lista creata in questo modo, è possibile liberare la memoria di ogni elemento, evitando cioè il *memory leak* (mancata deallocazione della *memoria heap*)?
- ### 2. SOMMA TRA LISTE LINEARI (soluzione in *list\_sum.c*):
- (a) scrivere un programma che crei due liste lineari di elementi omogenei *floating point* che occupano in memoria una posizione qualsiasi (non contigua).
- i. creare le due liste con un numero differente di elementi;
  - ii. inizializzare le due liste con numeri random nell'intervallo  $[0, 1)$ ;
  - iii. creare una terza lista che contenga la somma degli elementi delle precedenti liste. La lista somma avrà una lunghezza uguale alla minore delle altre due;
  - iv. visualizzare la terza lista.