

## Liste dublu inlantuite

Subiecte Laborator Marti 16-18

Avem urmatoarele structuri:

```
typedef struct node {  
    void * data;  
    struct node * next;  
    struct node * prev;  
} node;  
typedef struct TDoubleLinkedList {  
    node *head;  
    node *tail;  
    int len;  
} TDoubleLinkedList;
```

1. Pornind de la implementarea unei liste simplu inlantuite, sa se implementeze functiile pentru o lista dublu inlantuita (init, add, remove, free, print, print\_reversed). Pentru functia de add at position, trebuie avut in vedere daca se insereaza mai aproape de head sau de tail – daca se va insera mai aproape de tail parcurgerea listei trebuie facuta de la final la inceput. Testati pentru numere intregi si pentru stringuri.

2. Adaugati functia de add\_in\_order care sa insereze elemente in ordine – consideram elementele de tip int.

```
add_in_order(TDoubleLinkedList *list, void * data)
```

3. Modificati lista astfel incat sa utilizeze o santinela – nodurile head si tail din implementare vor fi inlocuite de nodul santinela, node\* sentinel. Sentinel → next va da capul listei (head), iar sentinel → prev va fi ultimul nod (tail). Daca nu exista niciun nod, santinela va pointa la ea insasi. De asemenea, primul nod va pointa la santinela in campul de prev, iar ultimul va pointa tot la santinela in campul de next.

Sentinel → Node1 → Node2 ... → Node n → Sentinel

4. Adaugati o functie care sorteaza o lista dublu inlantuita.