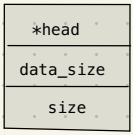
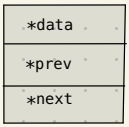
LinkedList Lab3

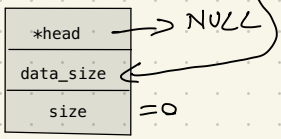
doubly_linked_list_t



dll_node_t

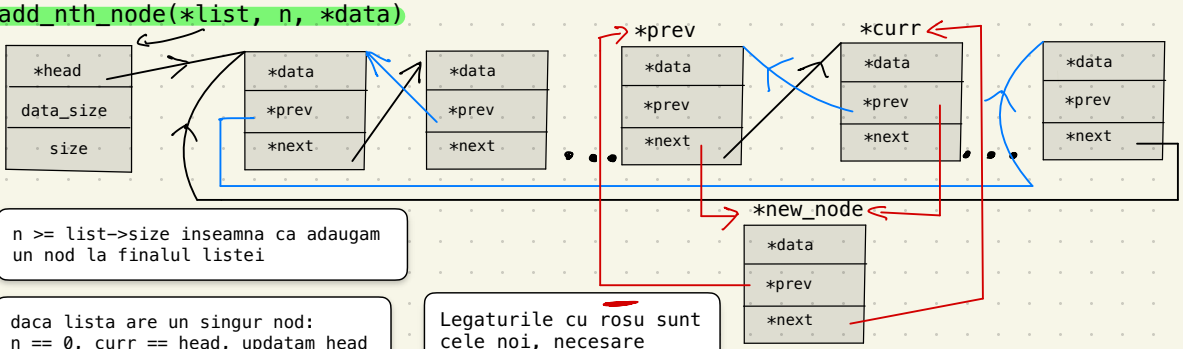


dll_create(data_size)



doar alocam structura de doubly_linked_list

add_nth_node(*list, n, *data)

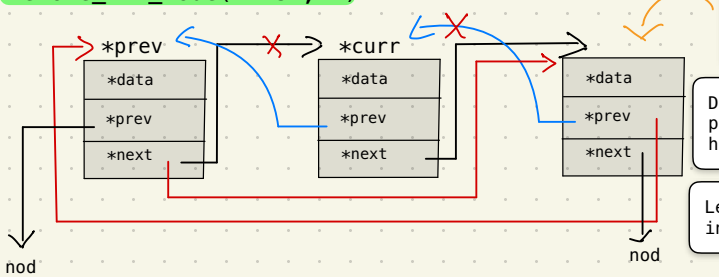


n >= list->size inseamna ca adaugam un nod la finalul listei

daca lista are un singur nod:
n == 0, curr == head, updatam head
n >= 1, nu updatam head

Legaturile cu rosu sunt cele noi, necesare adaugarii unui nou nod

remove_nth_node(*list, n)



Daca n==0, head va fi curr->next, in acest context, prev va reprezenta ultimul nod al listei circulare.

Daca dupa “parcurgere” pana la al n-lea nod, prev == curr, exista un singur element, adica head si il invalidam

Legaturile marcate cu x vor fi invalidate, inlocuite de cele noi cu rosu

dll_free(**pp_list)

daca pp_list sau *pp_list este NULL, nothing to do, return

```
Cat timp avem elemente in lista
     node = remove_nth_node(*pp_list, 0)
     eliberam memoria pt node->data
     eliberam memoria pt node
```

Eliberam memoria pt *pp_list si il setam pe NULL.