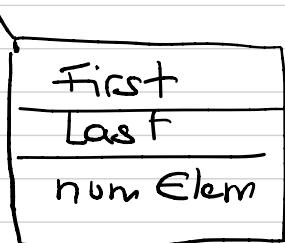


# INFO2 - Listes Chainées

9.VI.2020

Liste chainée



`typedef struct f`

`elem * first;`  
`elem * last;`  
`uint32_t numElem`

`↳ linkedList;`

`typedef struct elem {`

`struct elem * next;`  
 `struct elem * prev;`  
 `payload p;`  
`} elem;`

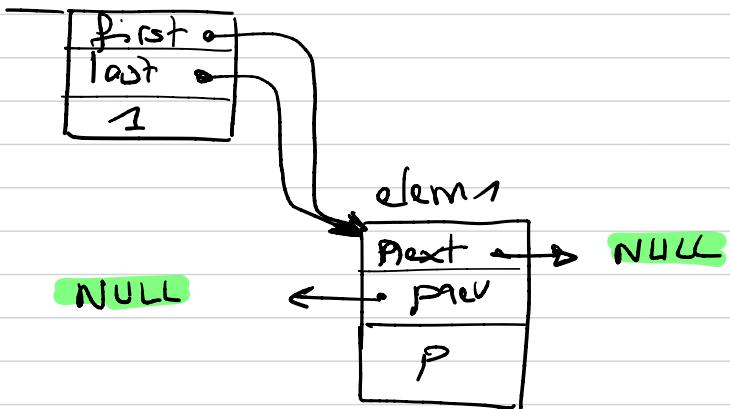
Liste vide =

`first = last = NULL`  
`numElem = 0.`

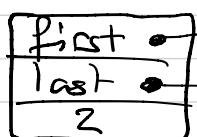
1 élément

e

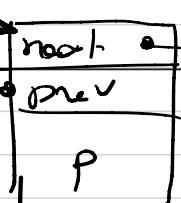
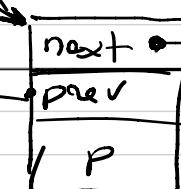
`first = last = 1 elem`  
`numElem = 1`



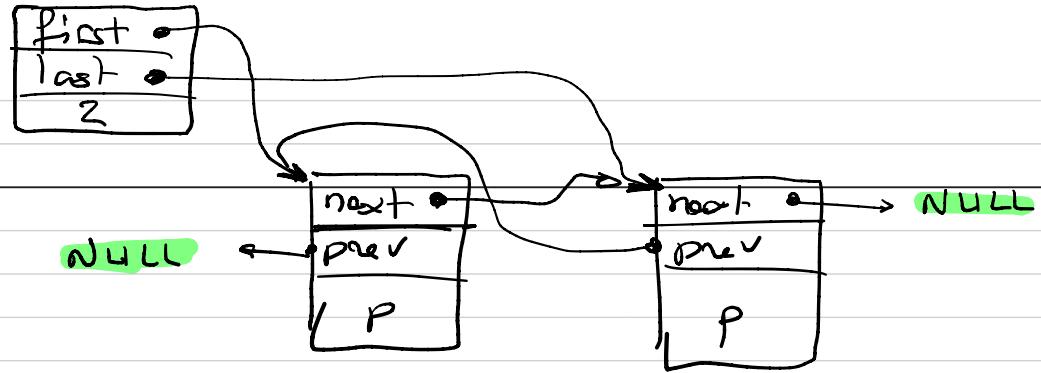
2 éléments



NULL



2 éléments



Insertion d'un élément

→ pos

"au début"

"à la fin"

"position spécifiée"

pos

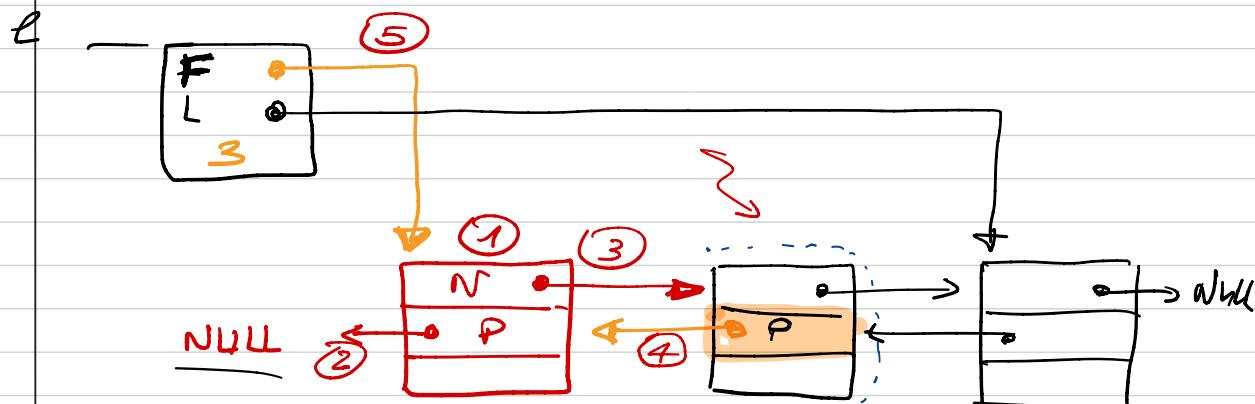
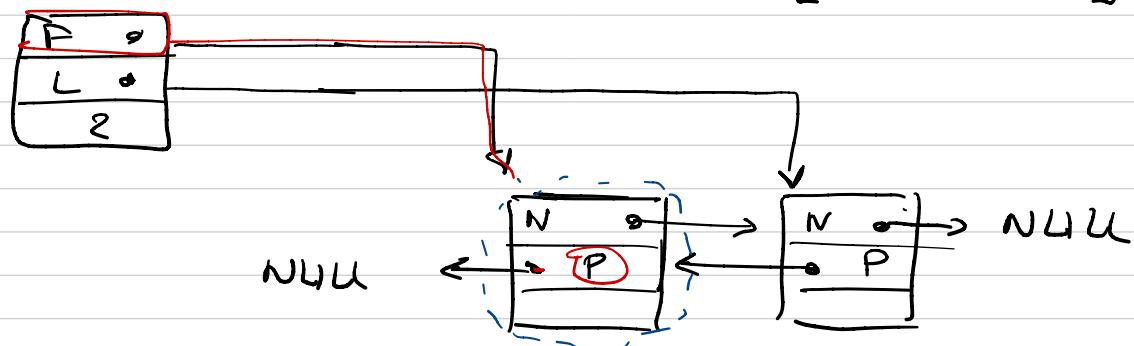
$\emptyset$

-1

n

$\in [0, \text{numElem}-1]$

$\text{Pos} = \emptyset$



1) Crédation du nouvel élément e

2)  $e \rightarrow \text{prev} = \text{NULL}$

3)  $e \rightarrow \text{next} = e \rightarrow \text{first}$

4)  $e \rightarrow \text{first} \rightarrow \text{prev} = e$

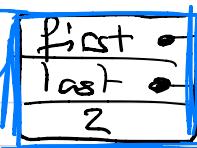
5)  $e \rightarrow \text{first} = e$

TD 2020 0602b  $\rightarrow$  insertElement (&L, &e, pos)

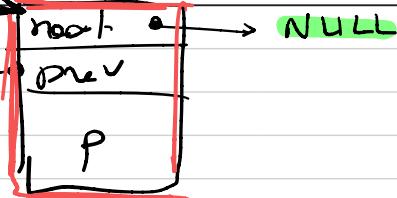
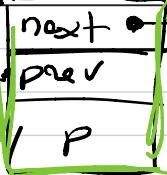
$\rightarrow$  displayList

~~ISAS~~  
~~15 b SS~~

pos =  $\emptyset$ .



NULL



2<sup>o</sup>-element

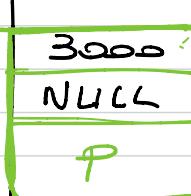
root  
prev



3000

1<sup>er</sup> element

root  
prev



2000

First  
last  
numElem

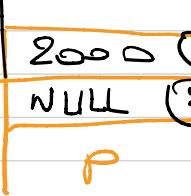


1000

1<sup>er</sup> element

① {

next  
prev



4000

2<sup>o</sup>-element  
3<sup>rd</sup>

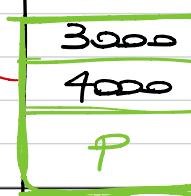
root  
prev



3000

1<sup>er</sup> element  
2<sup>nd</sup>

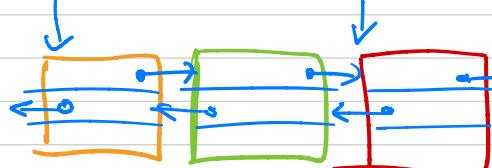
root  
prev



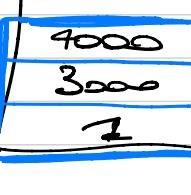
2000



NULL

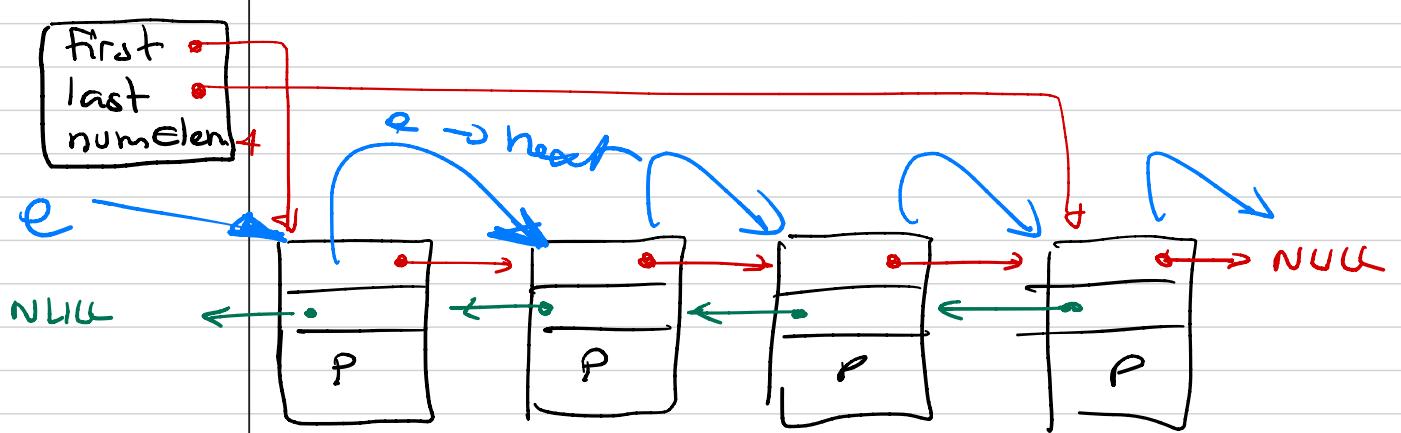


First  
last  
numElem



1000

## Display Elem.



- 1) accéder au premier
- 2) afficher le payload  $P$
- 3) aller au suivant
- 4) afficher le payload  $P$
- ;
- si suivant == NULL → stop

displayList ( linkedList \*l ) {

elem \*e = l -> first;

while

$\text{if} ( e \neq \text{NULL} ) \{$

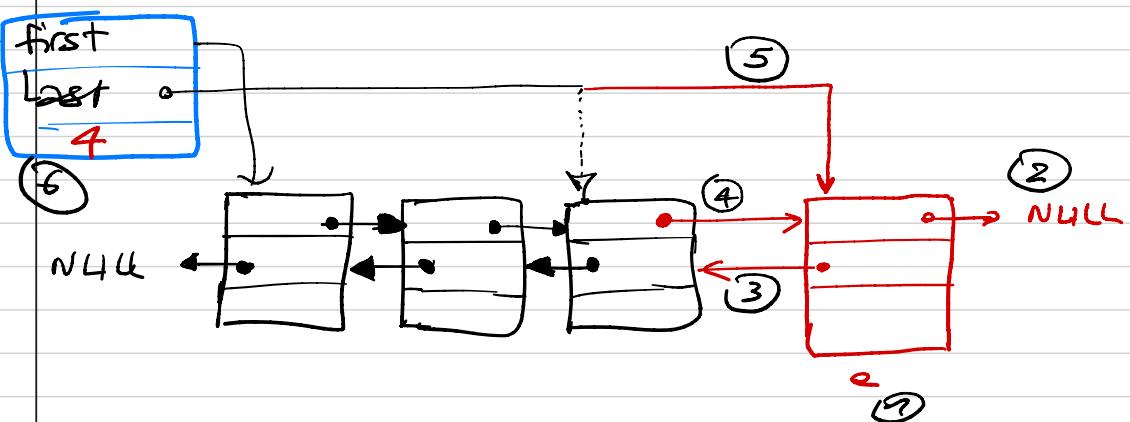
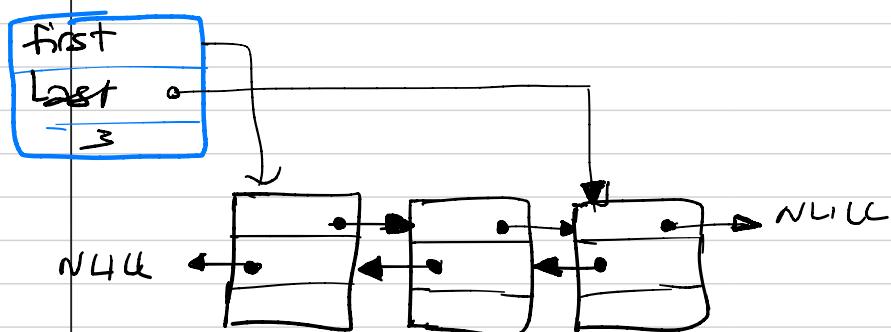
    displayElem ( e ); // ( , )

    e = e -> next;

}

Insert Elém position finale  
(->)

16'20



→ Ajoute le cas `pos = -1` dans `insertElém`

16'30