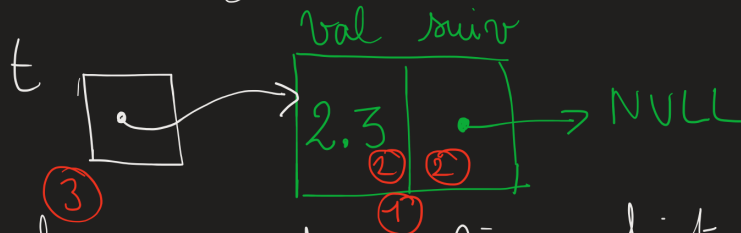


TOKEN  $t = \text{NULL}$ ;



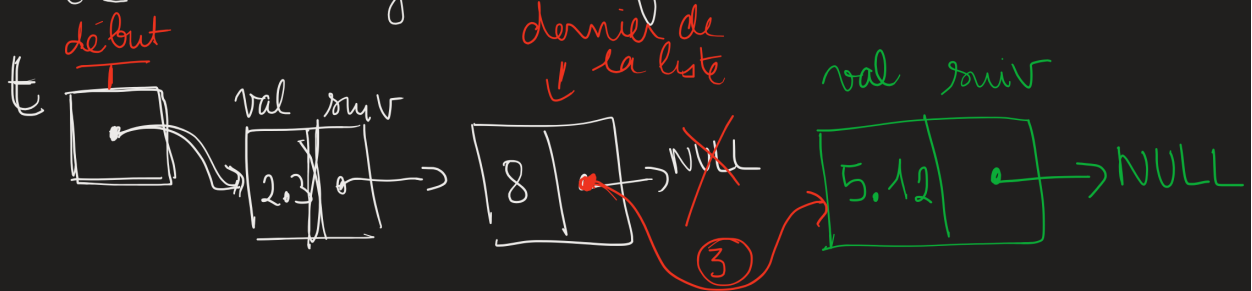
$t = \text{token\_ajouter\_fin\_liste}(t, 2.3);$



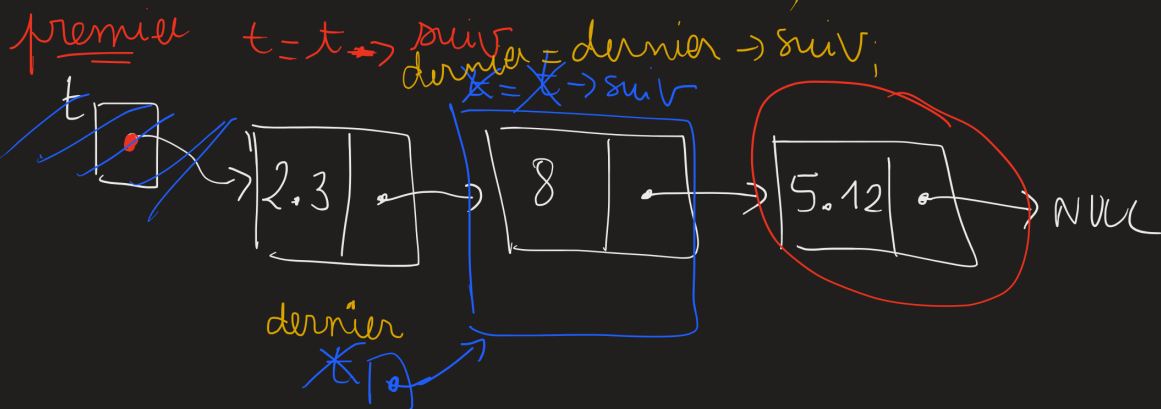
$t = \text{token\_ajouter\_fin\_liste}(t, 8);$

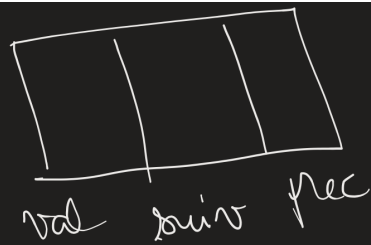


$t = \text{token\_ajouter\_fin\_liste}(t, 5.12);$



- ① Créer un nouveau TOKEN
  - ② Remplir avec les bonnes valeurs
  - ③ <sup>3.4</sup> Trouver le dernier élément de la liste  
Ajouter le TOKEN à la fin de la liste
  - ④ renvoyer la nouvelle liste
- $\text{dernier} = t;$

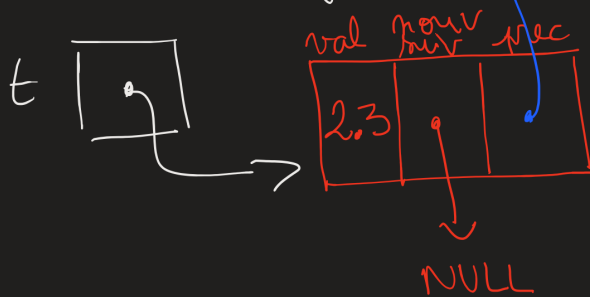




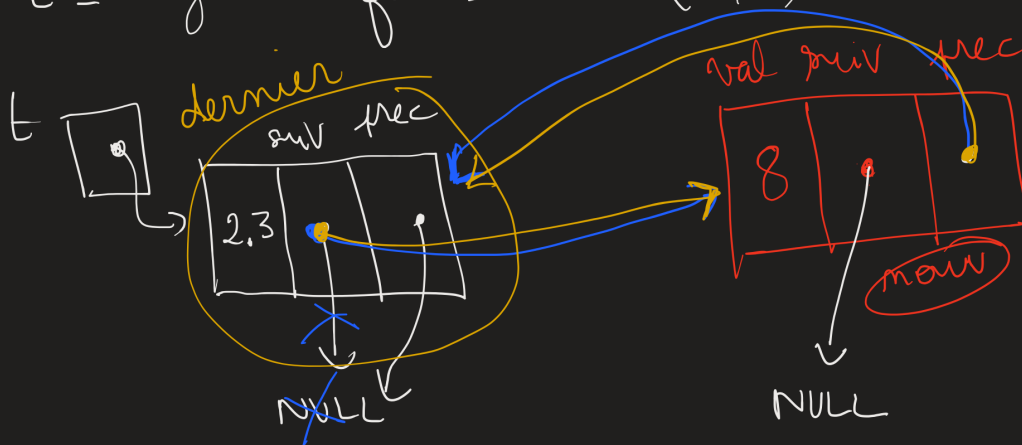
TOKEN  $t = \text{NULL}$ ;



$t = \text{ajouter\_fin\_liste}(t, 2.3);$



$t = \text{ajouter\_fin\_liste}(t, 8)$



- ① créer un nouvel élément
- ② mettre des valeurs dans le nouvel elt
- ③ chercher le dernier élément de la liste
- ④ le suivant du dernier  $\Rightarrow$  nouvel elt
- ⑤ le précédent du nouvel elt  $\Rightarrow$  dernier elt

op: + - / \*

( )

Valeurs 0 1 2 3 4 5 6 7 8 9

$(2 * (3 + 4))$

$\begin{cases} \text{val} : 2, 4, 2, 3 \\ \text{op} : (, ), *, -, \div, + \\ \text{type\_token} : \begin{cases} 0 : \text{op} \\ 1 : \text{val} \end{cases} \end{cases}$

① Modifier notre struct token

②  $(1 + 2)$

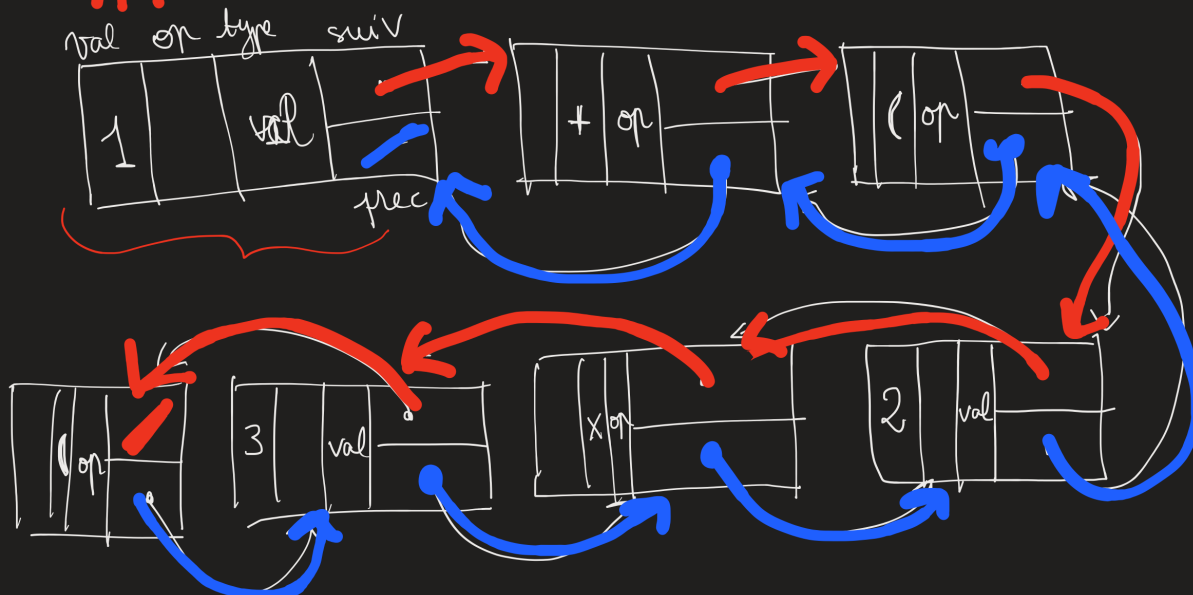
$1 + (2 * 3)$  A

$1 + (2 * 3)$

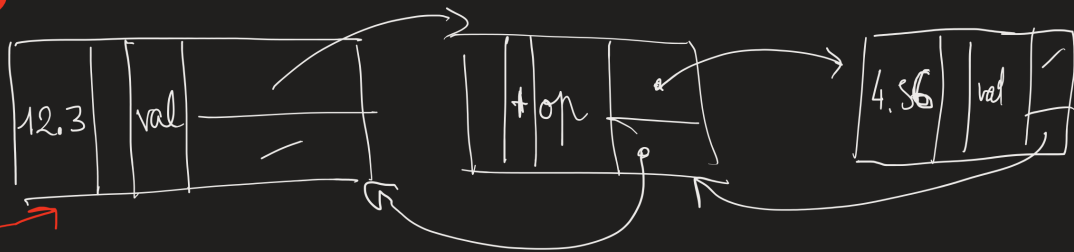
← règles de priorité

$12.3 + 4.56$  B

A:



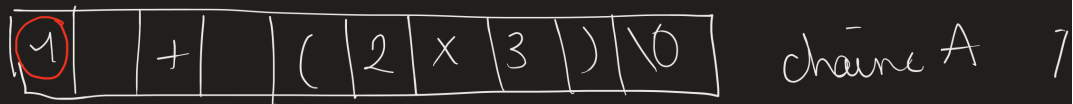
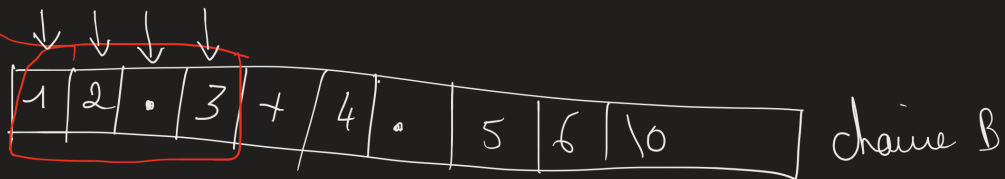
B



12.3 + 4.56

chaîne de caractères

1 + (2 x 3)



'1' →  $1 \times 10^0 = 1$   
 '2' →  $(1 \times 10^0) \times 10 + 2$

valeur avant  
la virgule  
V1

'.' → retenir la partie entière

'3' →  $3 \times 0,1 = 3/10$

'5' →  $(3 \times 0,1) + (5 \times 0,01) =$   
 '6' →  $5/100 + 6$

valeur  
après  
virgule  
V2

valeur = V1 + V2

'12.356' ← ch. de caractère

(12)(3)56 val réelle

4. val = 1

→  $1 \times 10 + 2 = 12 + 3/10$

$= 12,3 + 5/100$

$= 12,35 + 6/1000$

$= 12,356$