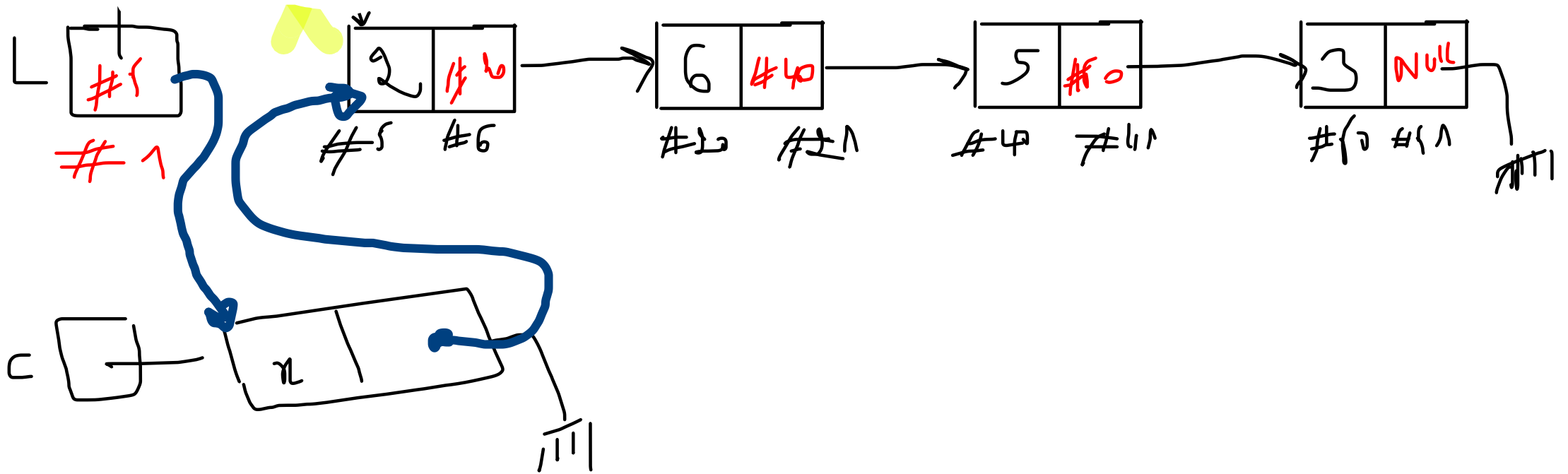


Chapitre 4. Liste simplement chaînée

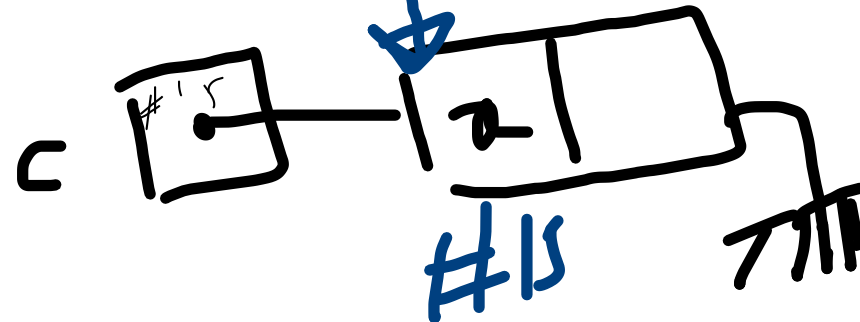
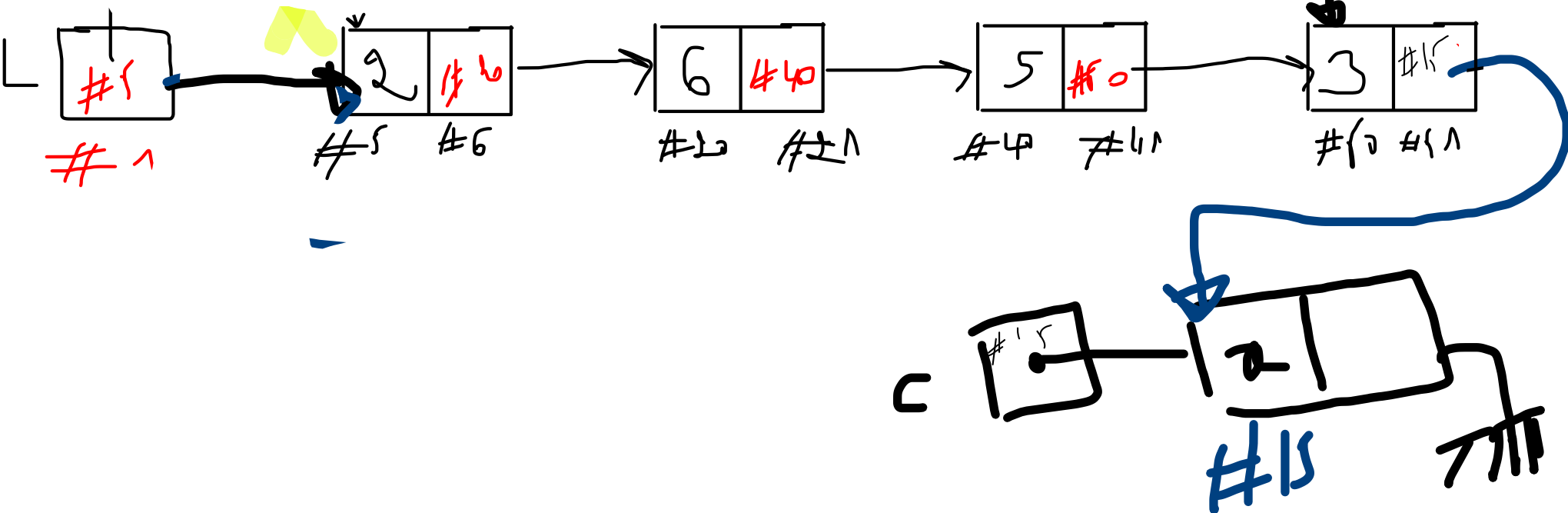
① A tout début

liste * L



$P \rightarrow \text{next} = \text{NULL}$

Ajout fin



$C = \#15$

$P \rightarrow \text{next} = C$

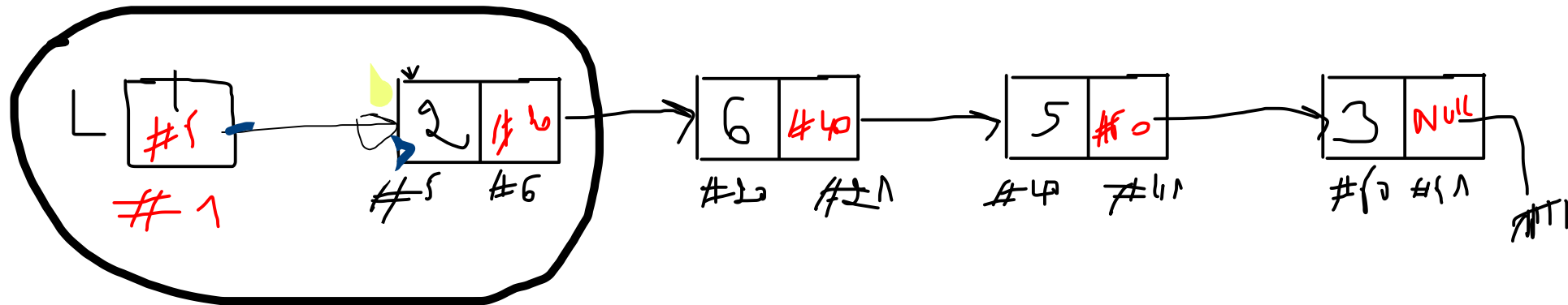
```
typedef struct cellule{
int info
struct cellule *suiv}liste
```

$L = \#5$

$L \rightarrow \text{info} = 2$

$\text{liste} = L$

$L \rightarrow \text{suiv} = \#20$



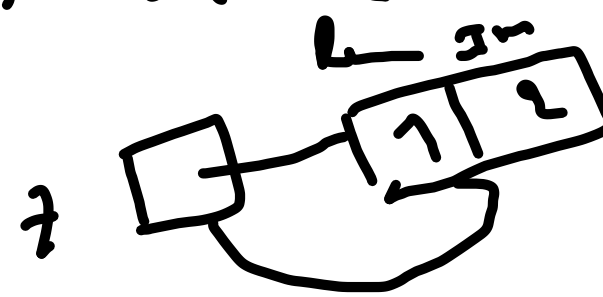
```
typedef struct complexe{
float re;
float im;}cplx;
```

$cplx * Z_i$

$Z = (cplx *) malloc(\text{---})$

$Z \rightarrow \text{Re}$

$Z \rightarrow \text{Im}$



$L = \#20$

$L \rightarrow \text{info} = 6$

$L \rightarrow \text{next} = 40$

$L = L \rightarrow \text{next}$

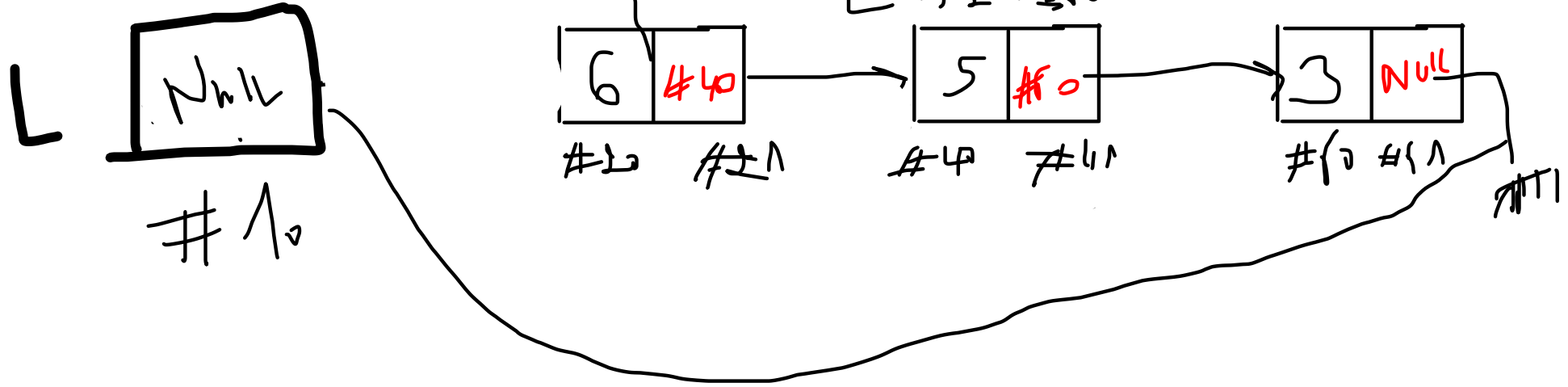
$L = \#40$

$L \rightarrow \text{info} = 5$
 $L \rightarrow \text{next} = 10$

$L = L \rightarrow \text{next}$

$L \rightarrow \text{info} = 3$

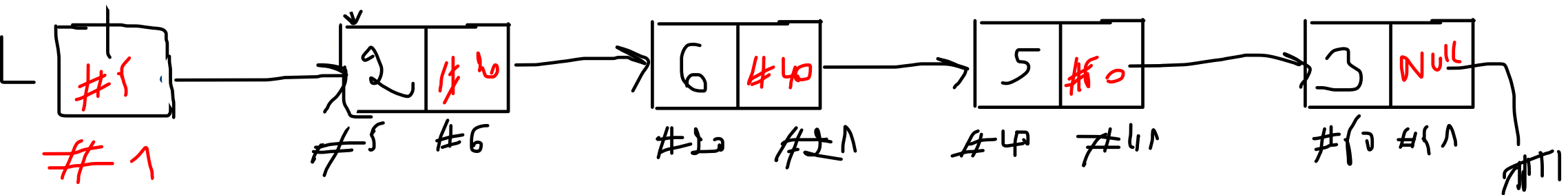
$L \rightarrow \text{next} = \text{Null}$



$L = L \rightarrow \text{next}$

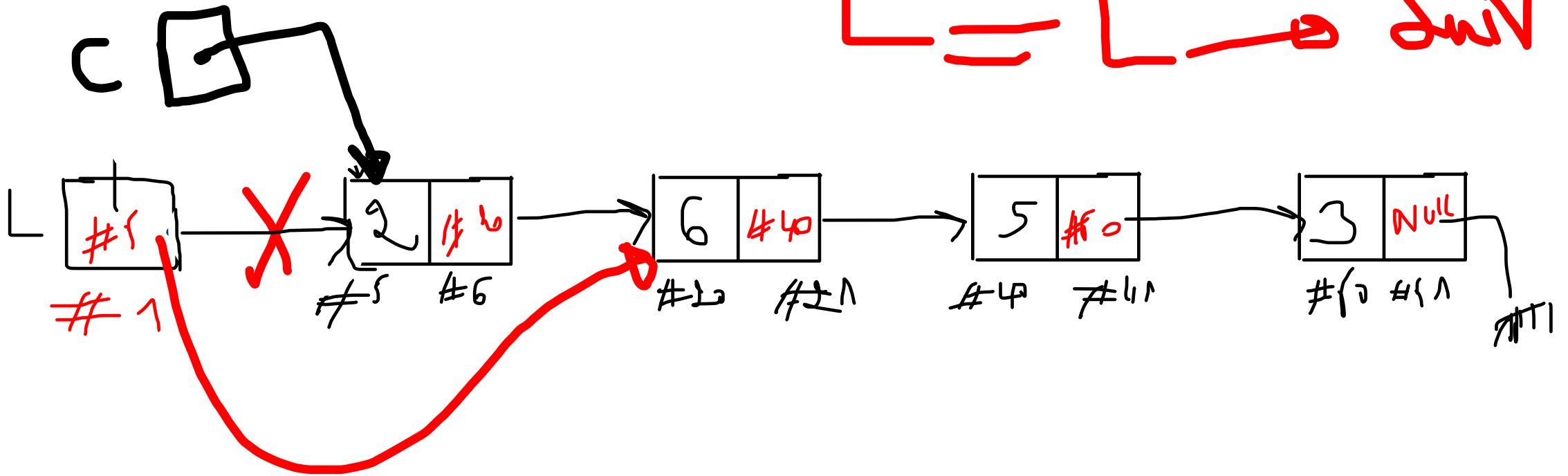
Supprimer un élément =

del A
Fin
x



Supprimer début

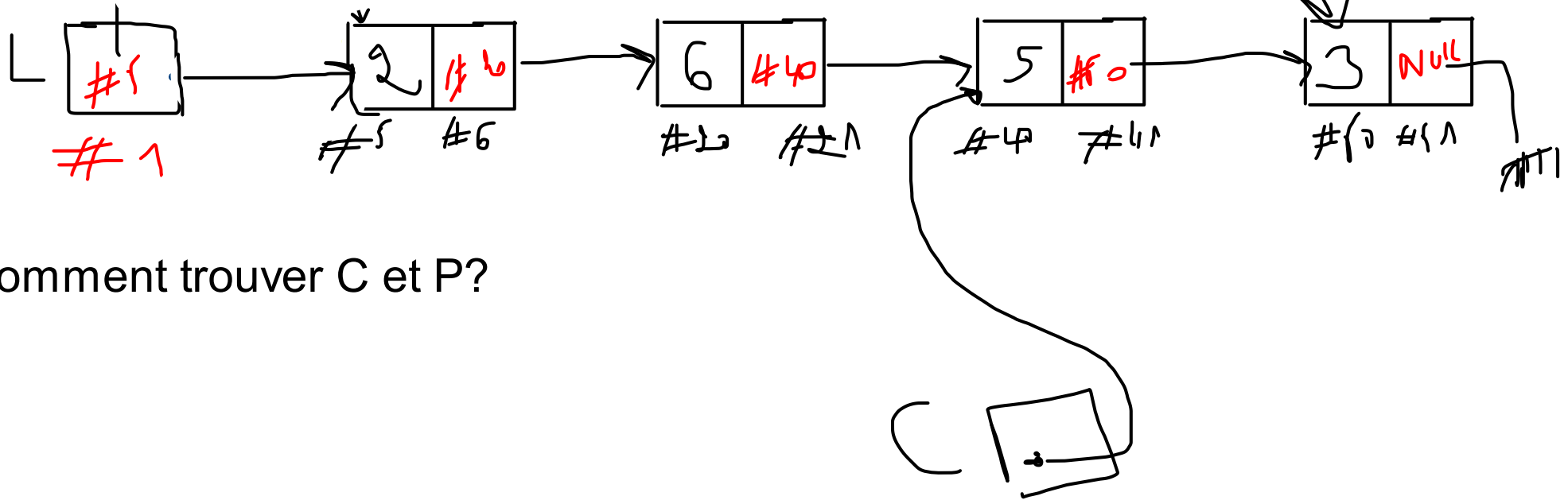
$L = L \rightarrow \text{Swil};$



liste * supprimerDebut(liste *L)

free(c)

Supprimer la fin



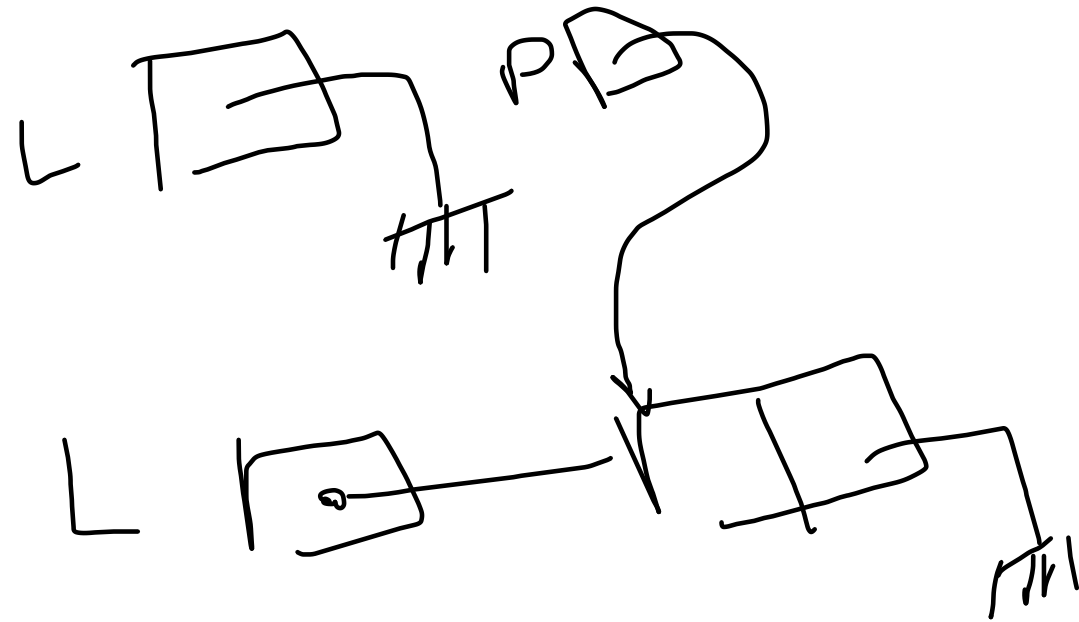
free (P)
C → NULL = NULL

Comment trouver C et P?

```

liste *SupprimerFin(liste *L)
{
    liste *c,*p;
    p=L;
    c=NULL;
    if (L==NULL)
        return L;
    while (p->suiv!=NULL)
    {
        c=p;
        p=p->suiv;
    }
    if(c==NULL)
        free(p)
    else
    {
        free(p)
        c->suiv=NULL;
    }
    return L;
}

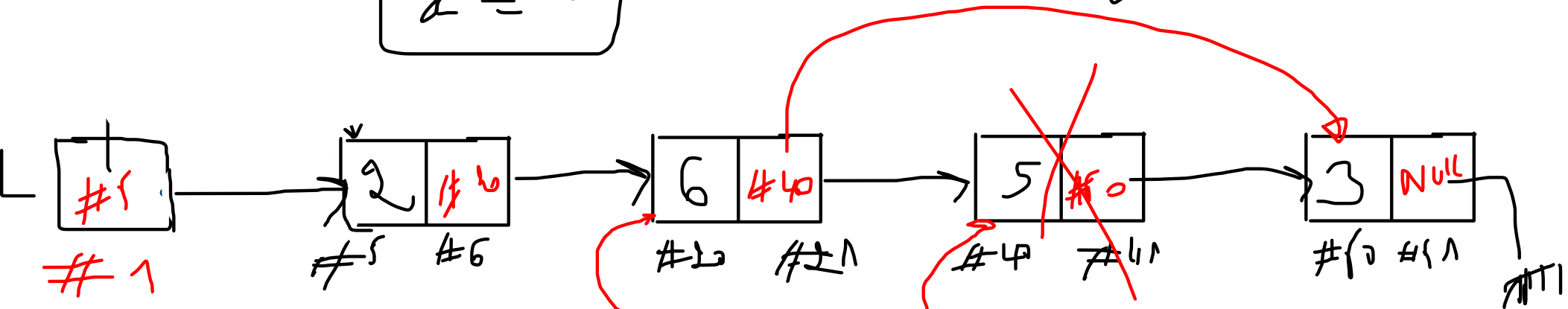
```



Supprimer élément x

$x = 5$

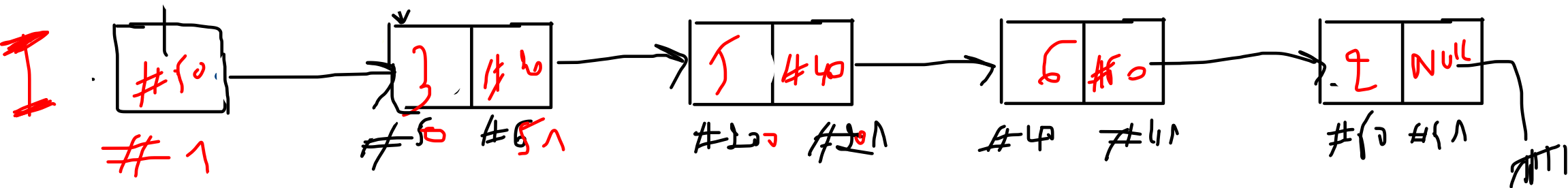
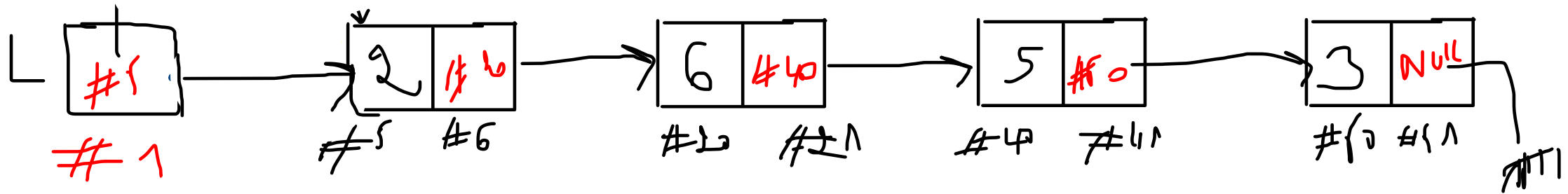
$x =$
 début
 Fin
 milieu
 n'existe pas cas //



liste * SupprimerElement(liste *L,int x)

$C \rightarrow \text{next} = P \rightarrow \text{next}$



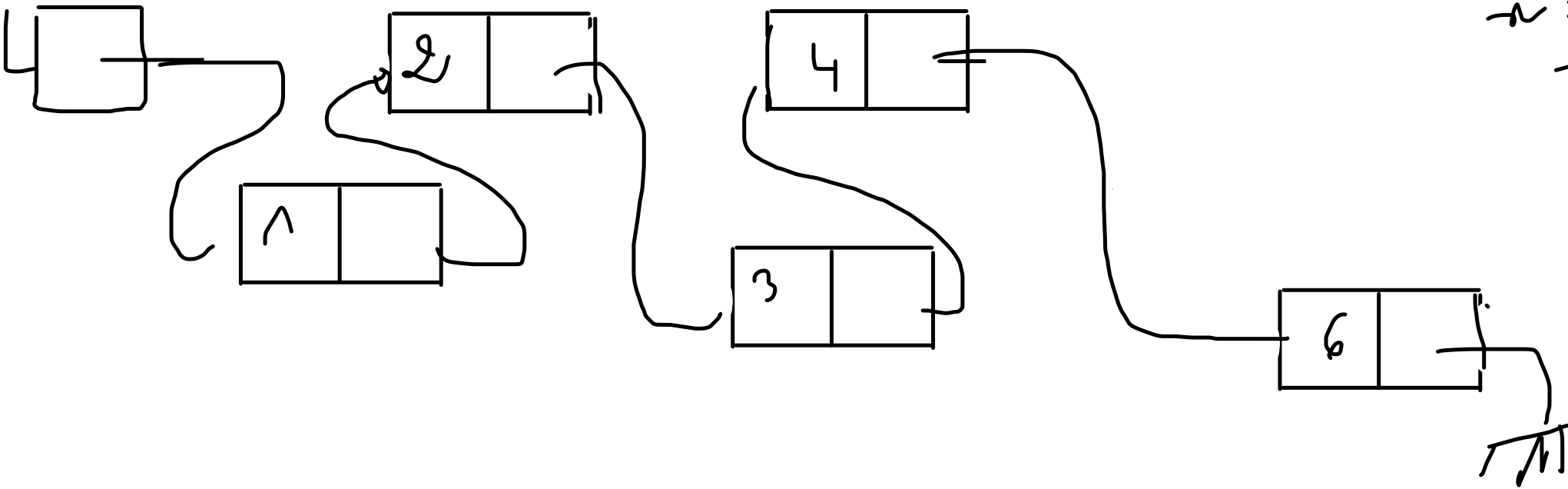


```
liste * InverserListe(liste *L)
```

```
{
    I=NULL
    while (L!=NULL)
    {
        I=InsertDebut(I,L->info);
        L=L->suiv;
    }
    return I ;
}
```

```
liste * InsertOrd(liste *L, int x)
```

$$\frac{L = \wedge}{n = 3}$$



$$\frac{n = 6}{}$$