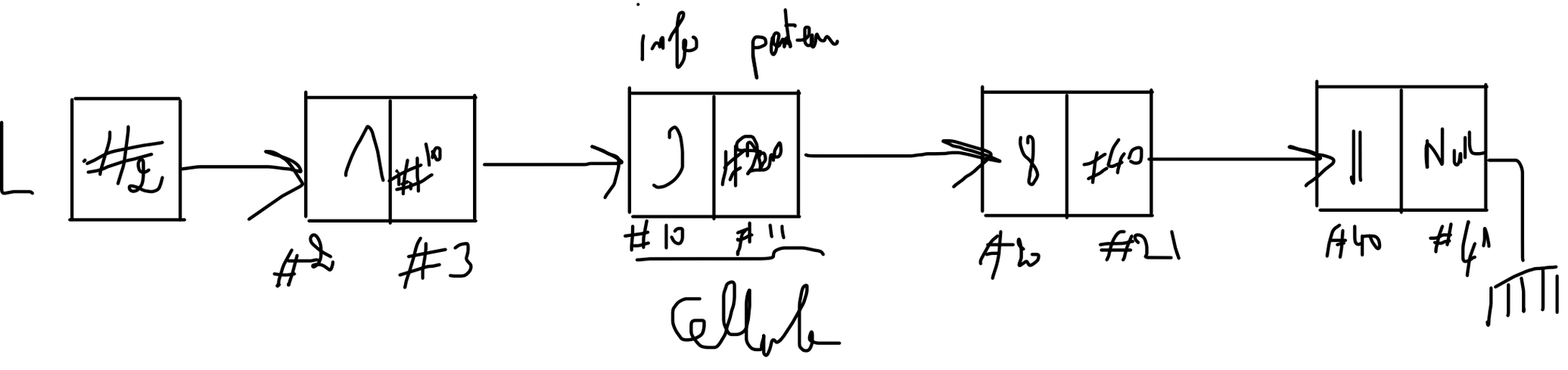
chapitre 4. liste simplement chaînée

Définition:

Une liste chaînée est une ensemble de cellules
Une cellule est une structure qui contient 2 champs (info et poiteur)



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

Une liste chaînée est un pointeur qui contient l'adresse de la première cellule

AjoutDebut(liste *L ,int x)

1. Déclarer la cellule
2. Créer la cellule
3. brancher

liste *C;

1

C=(liste*)malloc(sizeof(liste))

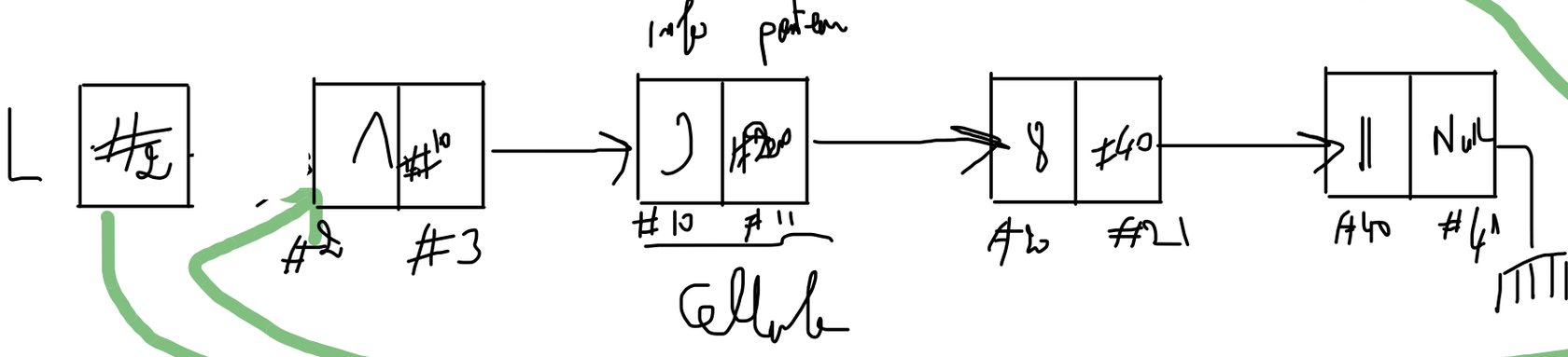
if (C!=NULL)

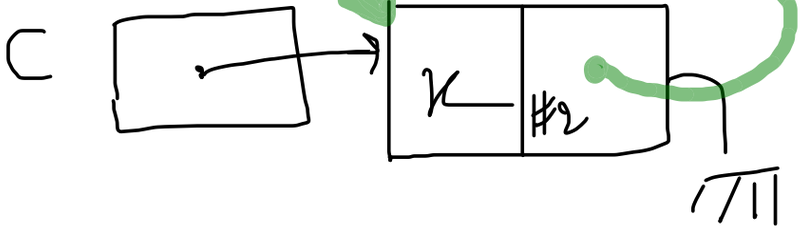
{C->info = x

C->suiv = NULL

}

2





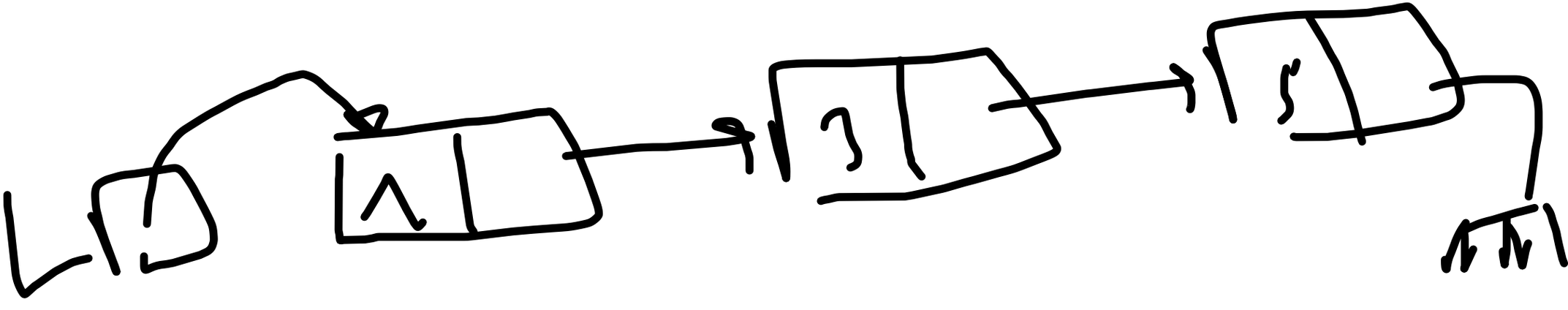
3

C->suiv = L

L = C

```c
////PP
main()
{
    liste *L=NULL;
    L=AjoutDebut(L,5);
    L=AjoutDebut(L,3);
    L=AjoutDebut(L,1);
    AfficherListe(L);
}
```

```
Sélection C:\Users\user\Desktop\S4_2022\listeChainee_S\LSC_V1.exe
1->3->5->NULL
--------------------------------
Process exited after 0.04686 seconds with return value 0
Appuyez sur une touche pour continuer...
```

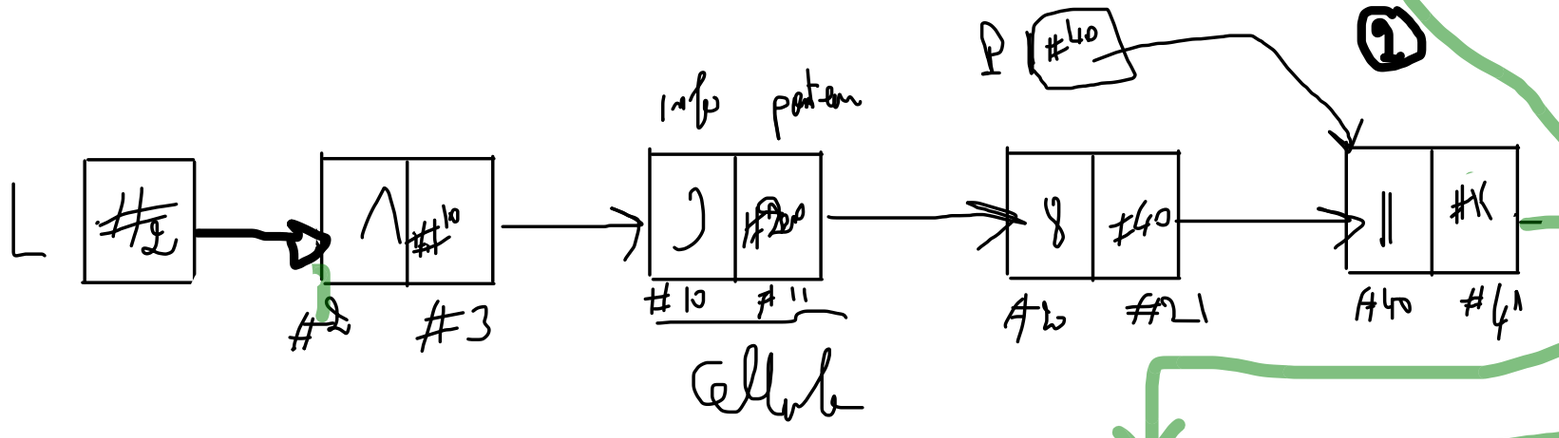


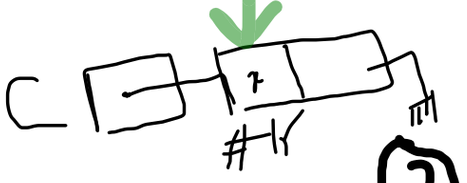
AjoutFin(liste *L ,int x)

1. Déclarer la cellule
2. créer la cellule
3. branchement

liste *C; (1)

C=(liste*)malloc(sizeof(liste))
if (C!=NULL)
{C->info=x
C->suiv=NULL
}

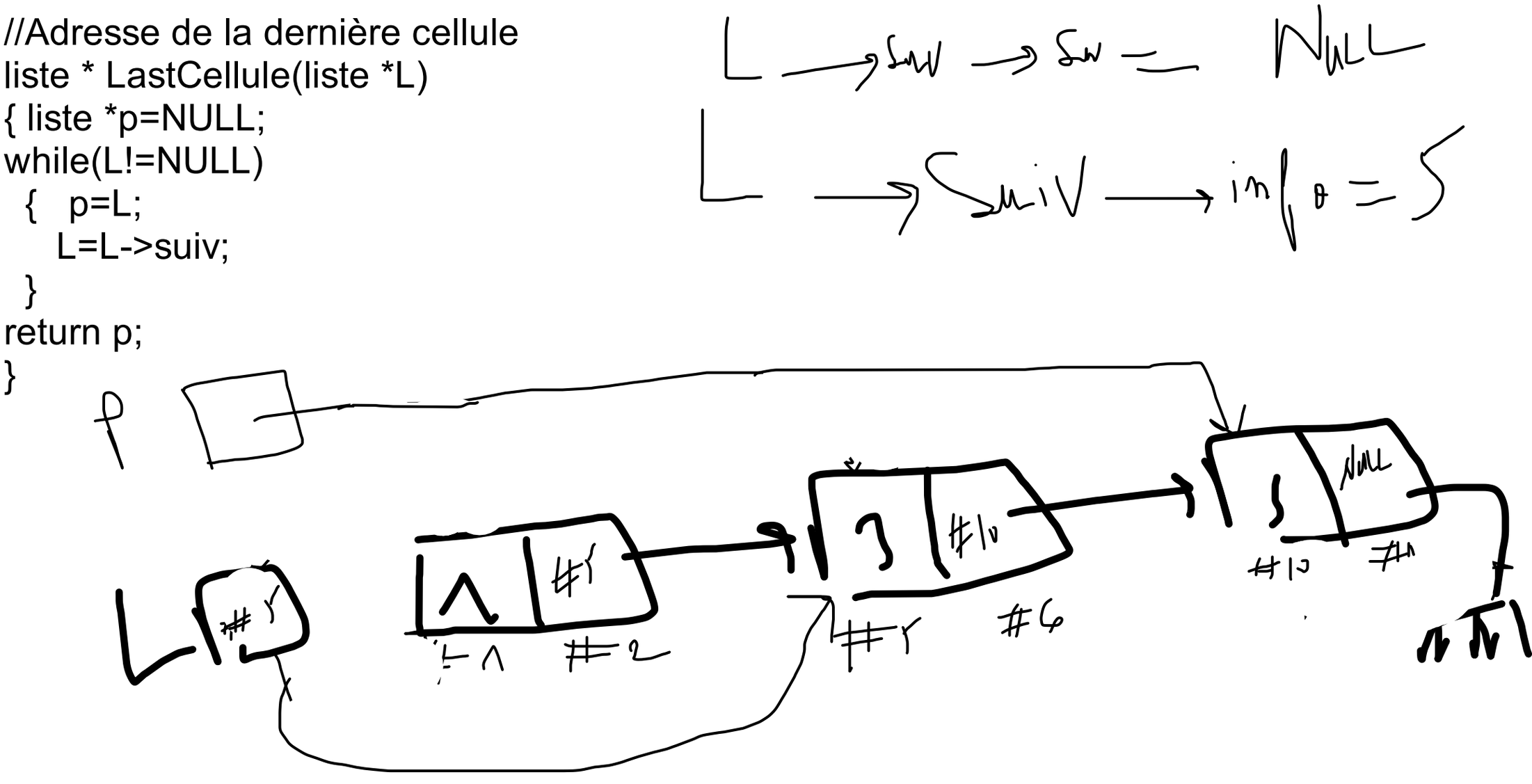




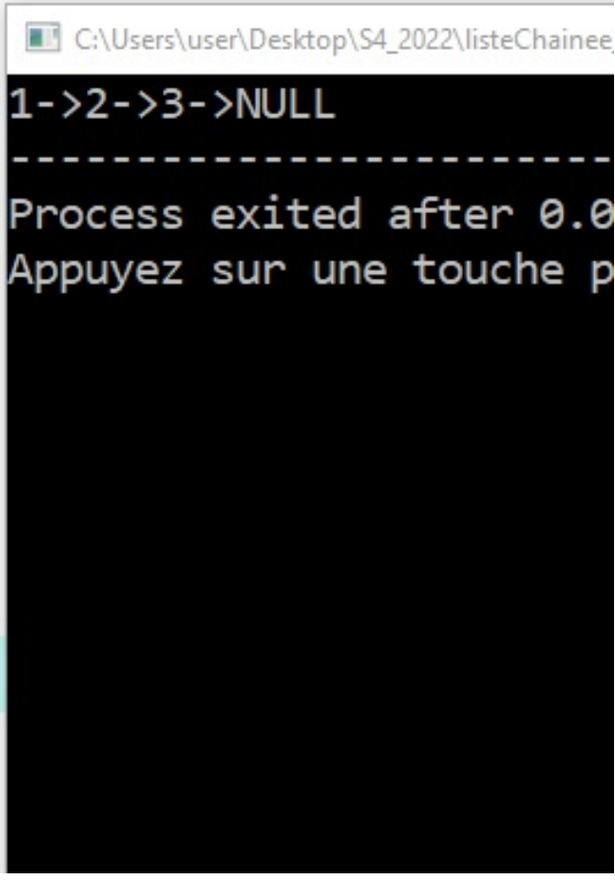
P est un pointeur qui contient l'adresse de la dernière cellule

P->suiv=C

```
//Adresse de la dernière cellule
liste * LastCellule(liste *L)
{ liste *p=NULL;
while(L!=NULL)
  {   p=L;
      L=L->suiv;
  }
return p;
}
```

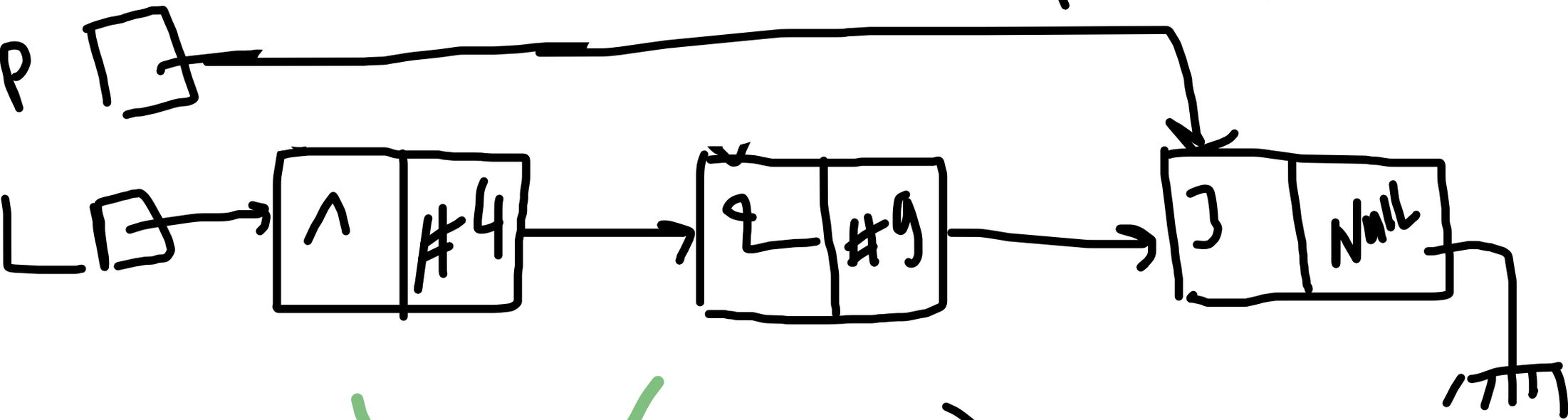


```c
////PP
main()
{
    liste *L=NULL;
    //L=AjoutDebut(L,5);
    //L=AjoutDebut(L,3);
    //L=AjoutDebut(L,1);
    L=AjoutFin(L,1);
    L=AjoutFin(L,2);
    L=AjoutFin(L,3);
    AfficherListe(L);
}
```



while (p -> next != NULL)

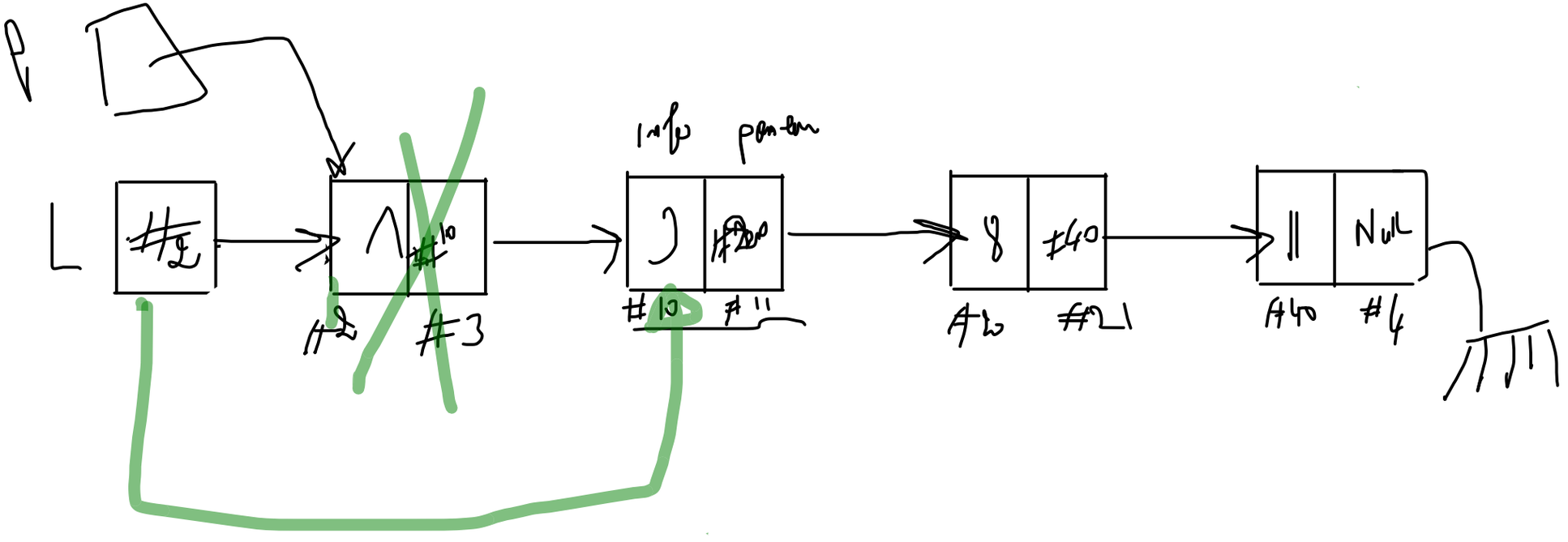
p = p -> next;



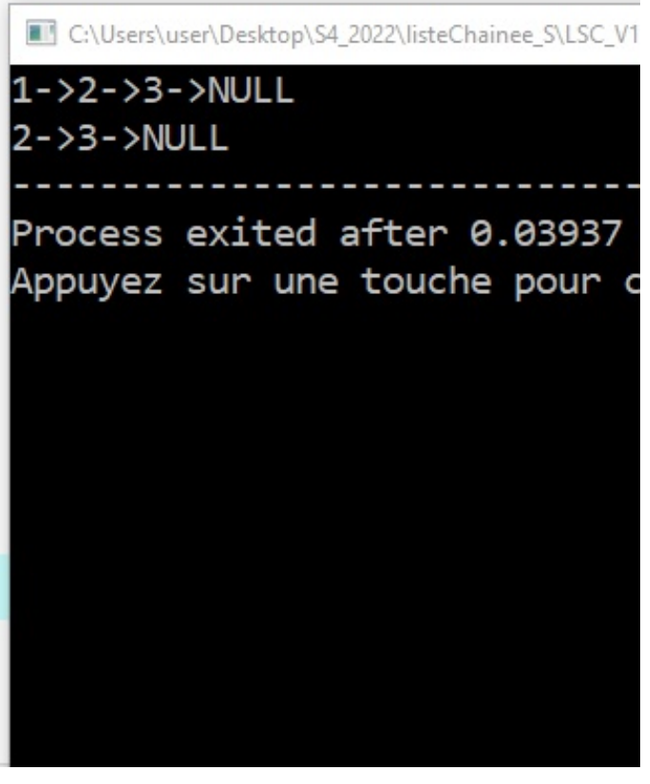
while (p != NULL)

p = p -> next;

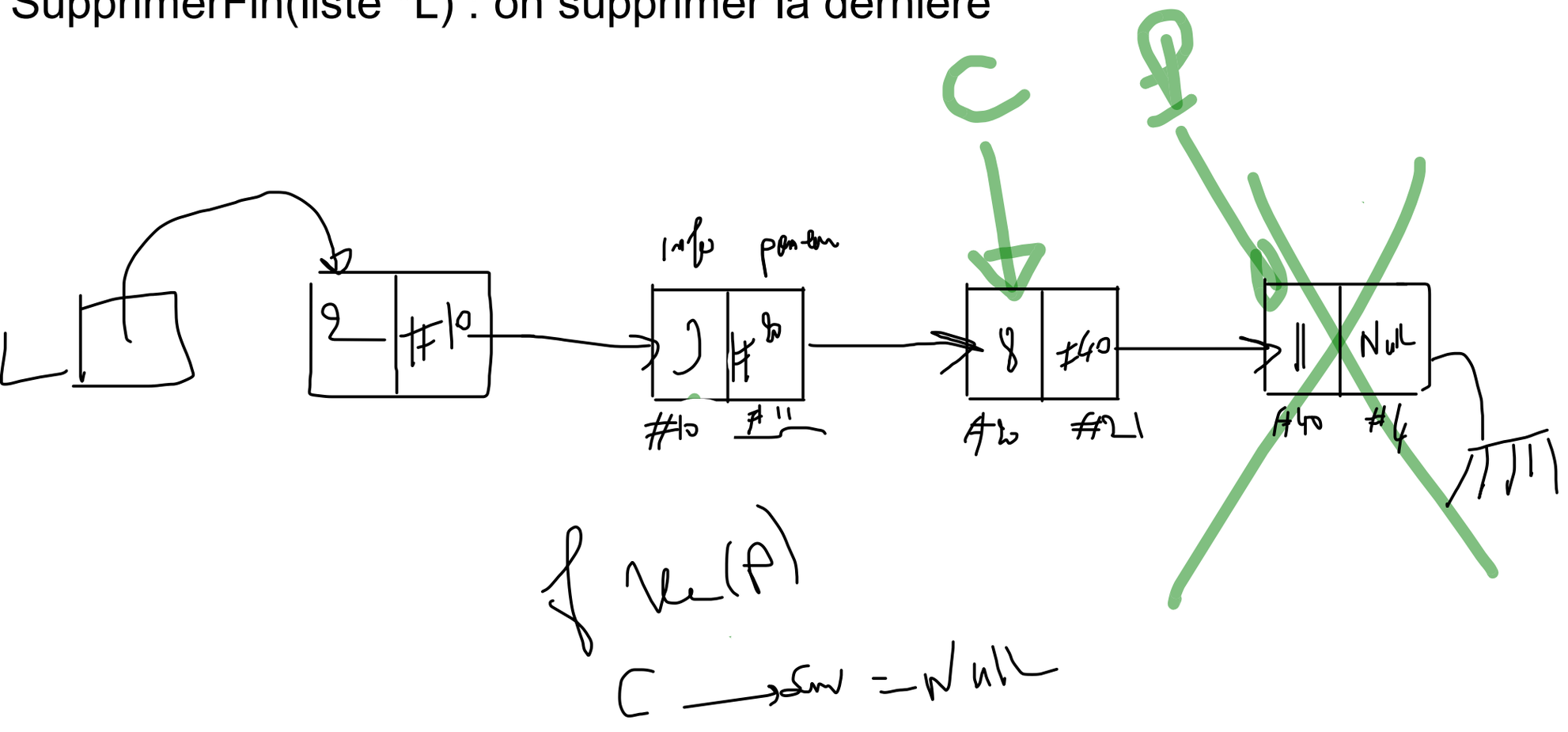
SupprimerDebut(liste *L) : on supprimer la première cellule



```c
////PP
main()
{
    liste *L=NULL;
    //L=AjoutDebut(L,5);
    //L=AjoutDebut(L,3);
    //L=AjoutDebut(L,1);
    //L=AjoutFin(L,1);
    //L=AjoutFin(L,2);
    //L=AjoutFin(L,3);
    L=AjoutFin1(L,1);
    L=AjoutFin1(L,2);
    L=AjoutFin1(L,3);
    AfficherListe(L);
    printf("\n");
    L=SupprimerDebut(L);
    AfficherListe(L);
```

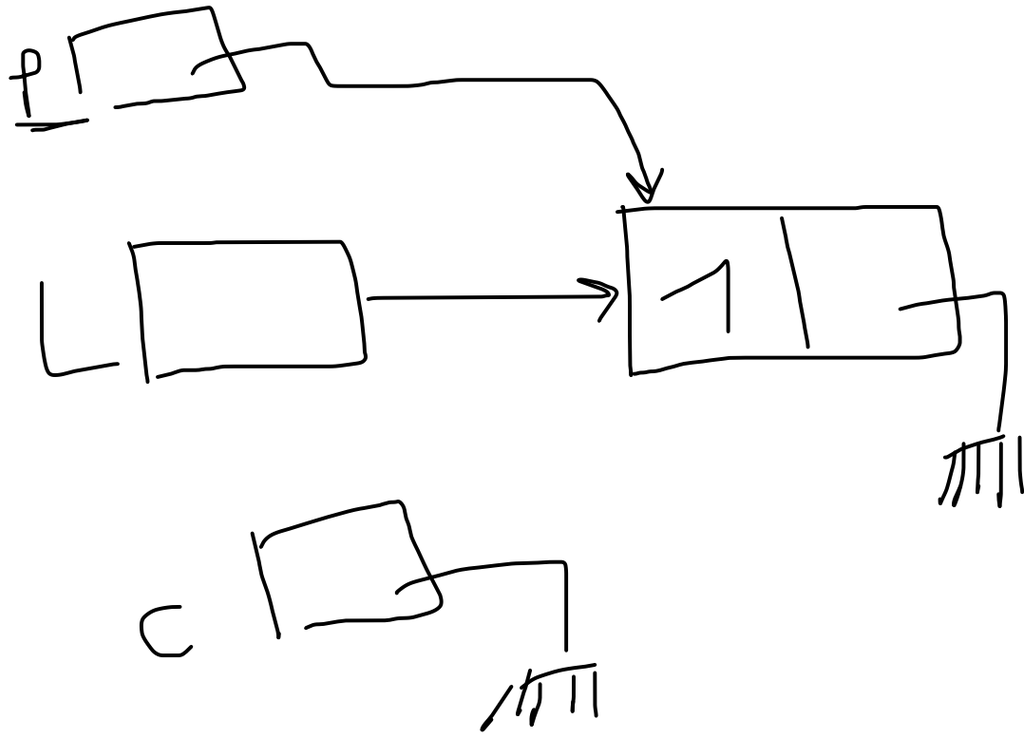


SupprimerFin(liste *L) : on supprimer la dernière

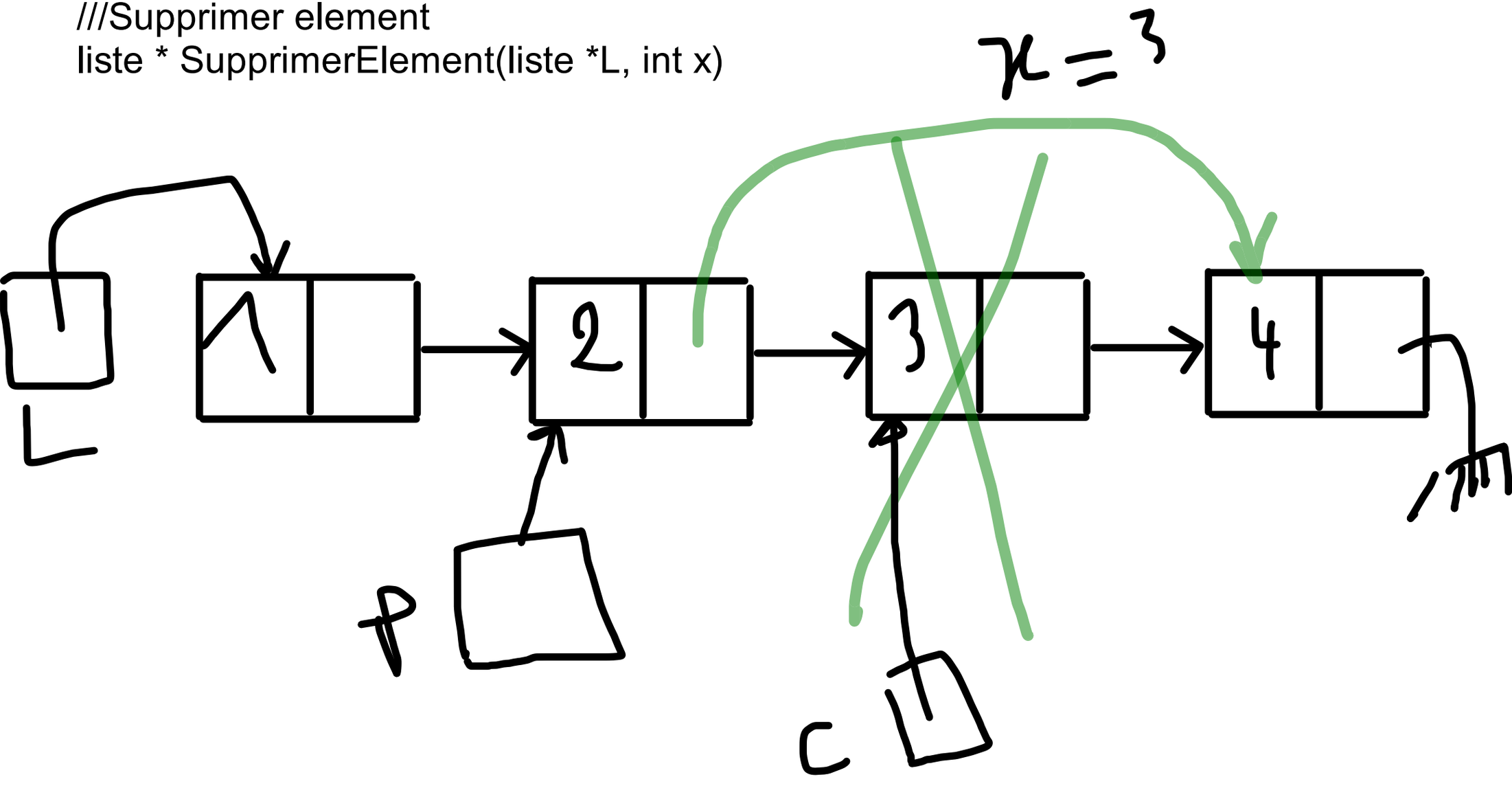


```c
/////////SupprimerFin
liste *SupprimerFin1(liste *L)
{   liste *c,*p;
    if (L!=NULL)
    {   p=L;
        c=NULL;
        //On cherche deux adresses:
        //c:adresse de la derniere cellule
        //p:adresse de l'avant derniere cellule
        while (p->suiv!=NULL)
        {
            c=p;
            p=p->suiv;
        }
        //la liste contient une seule cellule
        if(c==NULL)
                {
                free(p);
                L=NULL;
                }
        else
            {
                free(p);
                c->suiv=NULL;
            }
```

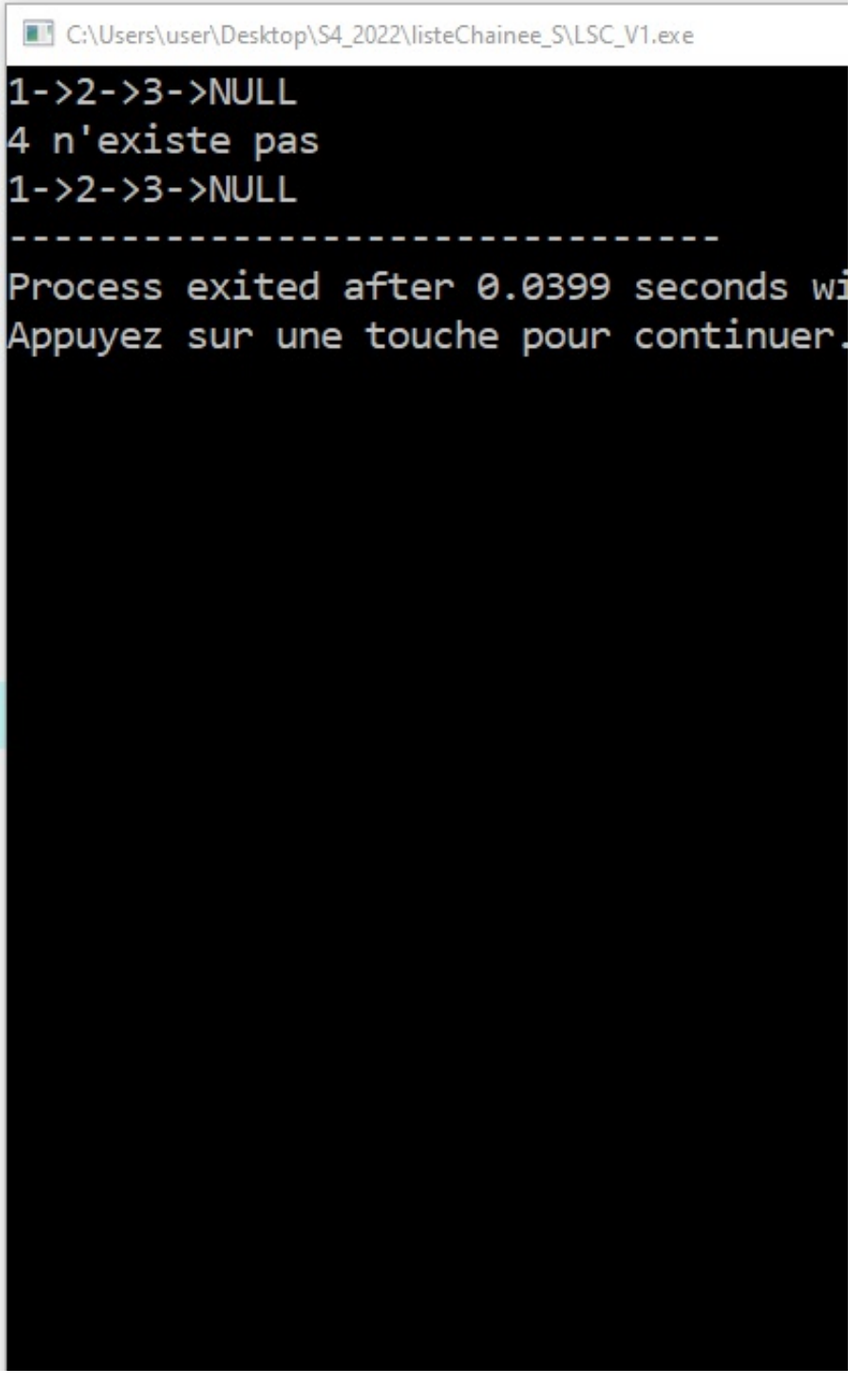
} return L; }

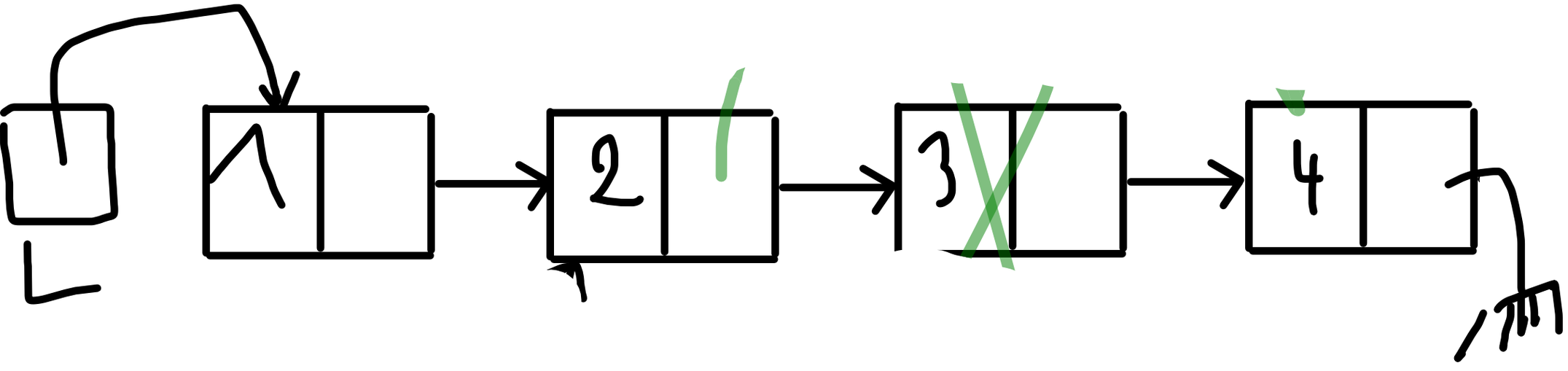


///Supprimer element
liste * SupprimerElement(liste *L, int x)

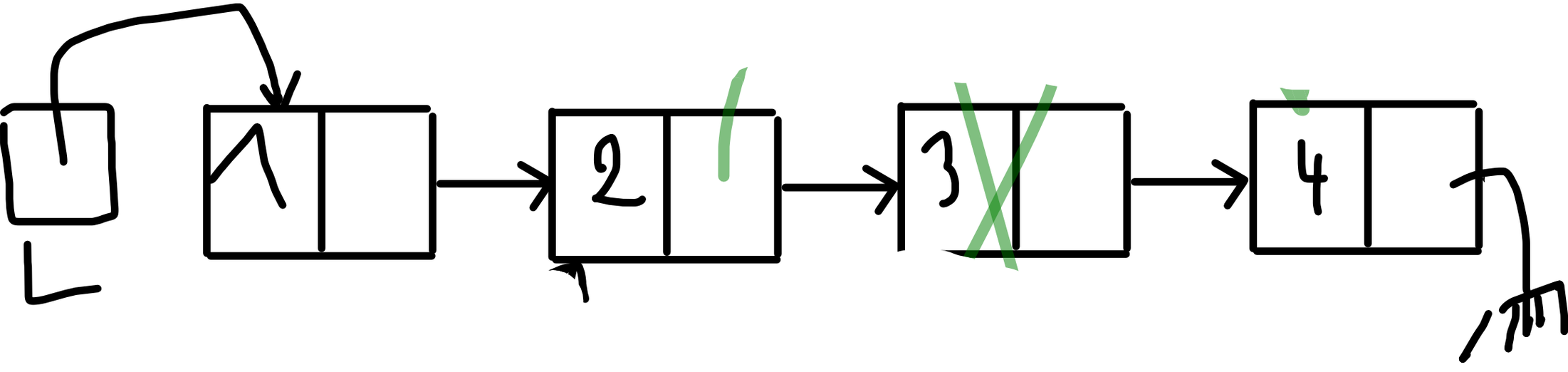


```
liste *L=NULL;
//L=AjoutDebut(L,5);
//L=AjoutDebut(L,3);
//L=AjoutDebut(L,1);
//L=AjoutFin(L,1);
//L=AjoutFin(L,2);
//L=AjoutFin(L,3);
L=AjoutFin1(L,1);
L=AjoutFin1(L,2);
L=AjoutFin1(L,3);
AfficherListe(L);
printf("\n");
L=SupprimerElement(L,4);
//L=SupprimerFin(L);
//L=SupprimerDebut(L);
//AfficherListe(L);
//printf("\n");
//L=SupprimerDebut(L);
//AfficherListe(L);
//printf("\n");
//L=SupprimerDebut(L);
AfficherListe(L);
```




1
2
3
4

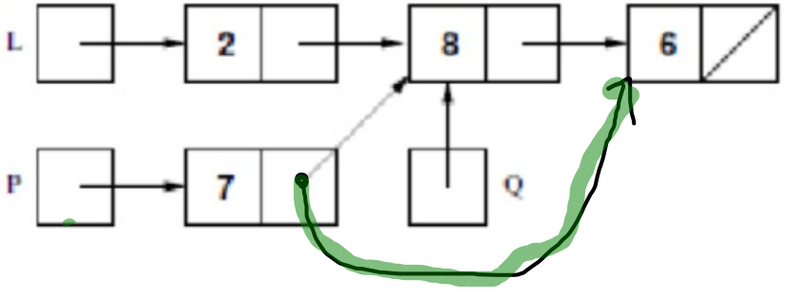
P = 1 x 2 x 3 x 4 = 24



```
typedef struct cellule
{
  int info;
  struct cellule *succ;
} liste;
```

Ouvrir avec Google Docs

On déclare trois listes L, P et Q : liste * L,*P,*Q.
La figure ci-dessous donne les représentations graphiques de 3 listes simplement chaînées L ; P ;Q.
Les séquences d'éléments de ces trois listes sont respectivement (2 ;8 ; 6) ; (7 ; 8 ; 6) et (8 ; 6).



Q2 L(6) P(7,8,6) Q(8,6)

Q1. On exécute l'instruction (Q− > succ)− > info = P− > info. Quelles sont les nouvelles séquences d'éléments des trois listes L ; P et Q ?

Q2. On revient à l'état décrit par la figure, puis exécute l'instruction L = Q− > succ. Quelles sont les séquences d'éléments de L ; P ;Q ?

Q3. On revient à l'état décrit par la figure, puis exécute l'instruction (L− > succ) = Q− > succ. Quelles sont les séquences d'éléments des listes L ; P ;Q ?

Q4. On revient à l'état décrit par la figure, puis exécute l'instruction P = P− > succ. Quelles sont les séquences d'éléments de L ; P ;Q ?

Q5. On revient à l'état décrit par la figure, puis exécute l'instruction P− > succ = Q− > succ. Quelles sont les séquences d'éléments de L ; P ;Q ?

Q1. L(2,8,7) P(7,8,7) Q(8,7)

Q2 L(6) P(7,8,6) Q(8,6)

Q3 L(2,6) P(7,8,6) Q(8,6)

Q4. L(2,8,6) P(8,6) Q(8,6)

Q5. L(2,8,6) P(7,6) Q(8,6)