Linked List – Devoir N°1

NB. les schémas doivent être dessinés à l'aide d'un logiciel

Exercice 1 :

Soit la structure de données suivante :

```
typedef struct element{
int data;
struct element *next;
}Element;
```

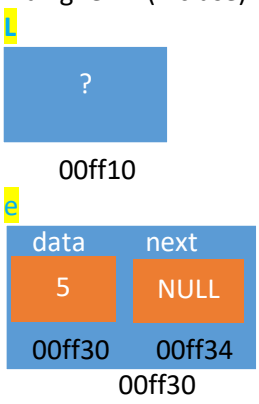
1)Comment expliquer que cette structure représente une structure d'une liste chainée

Dans le reste de l'énoncé, nous allons utiliser cette même structure de données.

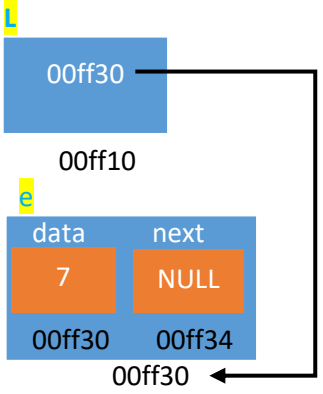
Exercice 1 : les constituant d'une liste simplement chainée

```
1   #include <stdio.h>
2   typedef struct element{
3   int data;
4   struct element *next;
5   }element;
6   int main()
7   {
8   Element *L;
9   Element e;
10  e.data=5;
11  e.next=NULL;
12  L=&e;
13  L->data=7;
14  return 0;
15  }
```

1) Donner le schéma de la mémoire jusqu'à la ligne 11 (incluse)



2) Donner le schéma obtenu jusqu'à la ligne 13 (incluse)

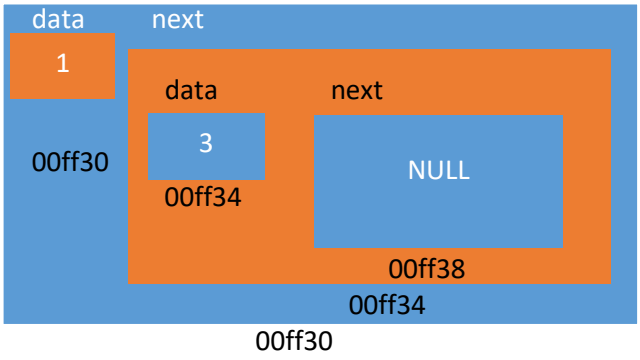


Exercice 2 : les constituant d'une liste simplement chainée

```
2  typedef struct element{
3  int data;
4  struct element *next;
5  }Element;
6  int main()
7  {
8  Element *L;
9  L=(Element*)malloc(sizeof(Element));
10 L->data=1;
11 L->next=(Element*)malloc(sizeof(Element));
12 L->next->data=3;
13 L->next->next=NULL;
14 Element *p;
15 for(p=L;p!=NULL;){
16     printf("%d - ",p->data);p=p->next;
17 }
18 return 0;
19 }
```

1) Donner le schéma de la mémoire jusqu'à la ligne 13 (incluse)

L



2) Donner le résultat du programme

Affichage : 1 – 3 –

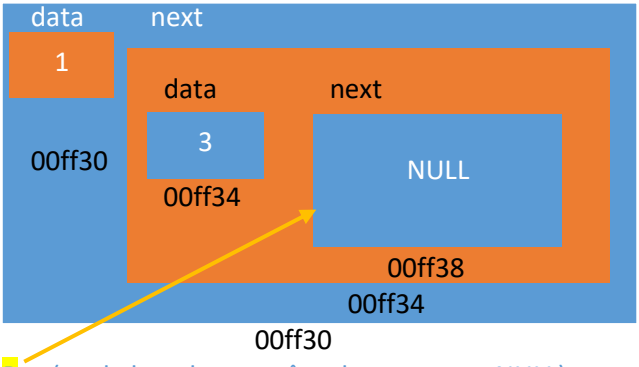
```
/*---PROGRAM1---*/
int main()
{
Element *L;
L=(Element*)malloc(sizeof(Element));
L->data=1;
L->next=(Element*)malloc(sizeof(Element));
L->next->data=3;
L->next->next=NULL;
Element *p;

for(p=L;p!=NULL;p=p->next); /*ligne 11*/
p->next=(Element*)malloc(sizeof(Element)); /*ligne 12*/
p->next->data=5;
p->next->next=NULL;

return 0;
}
```

1) Donner le schéma de la mémoire jusqu'à la ligne 11 (incluse)

L



P (car la boucle va arrêter lorsque p == NULL)

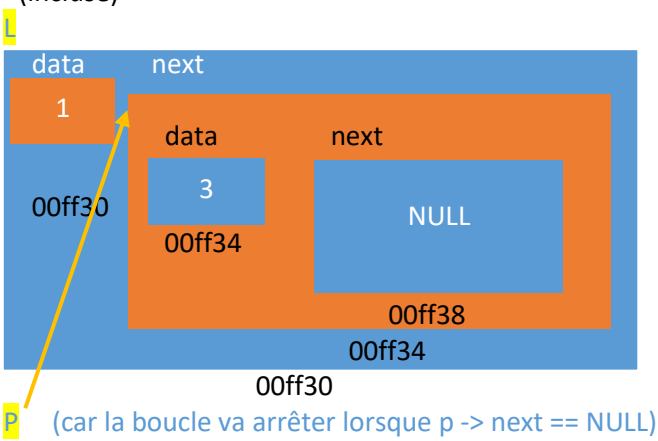
2) Pourquoi la ligne 12 contient un problème ? quelle est le résultat du programme ?

Le problème c'est que P est égal à NULL, et elle n'est pas pré-alloué. Le programme va exécuter avec erreur.

```
/*---PROGRAM2---*/
int main()
{
Element *L;
L=(Element*)malloc(sizeof(Element));
L->data=1;
L->next=(Element*)malloc(sizeof(Element));
L->next->data=3;
L->next->next=NULL;
Element *p;
for(p=L;p->next!=NULL;p=p->next); /*ligne 11*/
p->next=(Element*)malloc(sizeof(Element)); /*ligne 12*/
p->next->data=5;
p->next->next=NULL;

return 0;
}
```

3) Donner le schéma de la mémoire jusqu'à la ligne 11 (incluse)



(car la boucle va arrêter lorsque p -> next == NULL)

4) Pourquoi la ligne 12 ne contient pas un problème ? proposer un schéma qui montre les éléments de la liste

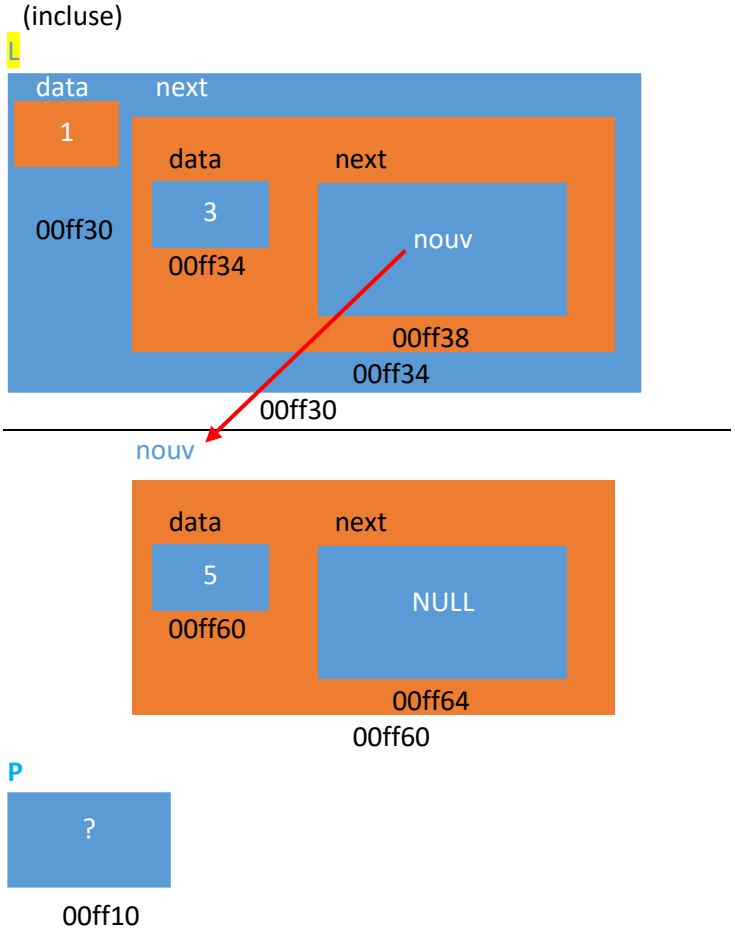
La ligne 12 ne contient pas de probléme, car on text si p -> next == NULL avant de se deplacer.


```
/*---PROGRAM3---*/
int main()
{
Element *L,*nouv,*p;
L=(Element*)malloc(sizeof(Element));
L->data=1;
L->next=(Element*)malloc(sizeof(Element));
L->next->data=3;
L->next->next=NULL;

nouv =(Element*)malloc(sizeof(Element));
nouv->data=5;
nouv->next=NULL;

for(;L->next!=NULL;L=L->next);
L->next=nouv; /*ligne 16*/
for(p=L;p!=NULL;p=p->next){
printf("%d - ",p->data);
}
return 0;
}
```

5) Donner le schéma de la mémoire jusqu'à la ligne 16 (incluse)



6) Quel est le résultat à afficher par le programme ?
Affichage : 3 – 5 –

```
/*---PROGRAM4---*/
int main()
{
Element *L,*elem;
elem=(Element*)malloc(sizeof(Element));
elem->data=1;
L=elem;

elem=(Element*)malloc(sizeof(Element));
elem->data=3;
L->next=elem;

elem=(Element*)malloc(sizeof(Element));
elem->data=4;
L->next->next=elem;

elem=(Element*)malloc(sizeof(Element));
elem->data=5;elem->next=NULL;
L->next->next->next=elem; /*ligne 13*/

for(elem=L;elem!=NULL;elem=elem->next){
printf("%d - ",elem->data);
}
/*calculer la somme des éléments*/
Int somme =0;
for(elem=L;elem!=NULL;elem=elem->next){
    somme += elem->data;
}
/*calculer la somme des éléments impairs*/
Int sommeImpaire =0;
for(elem=L;elem!=NULL;elem=elem->next){
  if(elem->data % 2 != 0)
    sommeImpaire += elem->data;
}
return 0;
}
```

7) Donner le schéma de la mémoire jusqu'à la ligne 13 (incluse)

L

8) Quel est le résultat à afficher par le programme ?
Affichage : 1 – 3 – 4 – 5 –

9) Écrire le code pour calculer la somme des éléments de la liste

10) Écrire le code pour calculer la somme des éléments impairs