

```

void Print_List (struct node *n) {
    while (n != NULL) {
        printf("%d", n->data);
        n = n->next;
    }
    Print_List (head);
    return 0;
}

```

⊗ Initialiser un liste:

```

Node *initialiser() {
    return NULL;
}

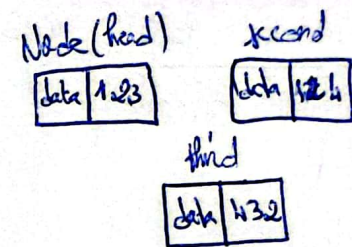
void main() {
    Node *ptr Node;
    ptr Node = initialiser();
}

```

```

int main() {
    struct Node {
        int data;
        struct *Node *next;
    };
}

```



```

struct Node * head = NULL;
struct Node * second = NULL;
struct Node * third = NULL;

```

```

head = (struct Node*) malloc (sizeof(struct Node));
second = (struct Node*) malloc (sizeof(struct Node));
third = (struct Node*) malloc (sizeof(struct Node));

head->data = 1;
head->next = second;
second->data = 2;
second->next = third;
third->data = 3;
third->next = NULL;

```

⊗ Création d'une liste

```
typedef struct cellule {
```

```
    int val;
```

```
    cellule *suiv;
```

```
} cellule;
```

```
cellule *creation(int n){
```

```
    cellule *p, *tete, *q;
```

```
    int x, i;
```

```
    p = malloc(sizeof(cellule));
```

```
    scanf("%d", &x);
```

```
    p->val = x;
```

```
    p->suiv = NULL;
```

```
    tete = p;
```

```
    for(i=0; i<n; i++){
```

```
        q = p;
```

```
        p = malloc(sizeof(cellule));
```

```
        scanf("%d", &x);
```

```
        p->val = x;
```

```
        p->suiv = NULL; q->suiv = p;
```

```
    return tete;
```

(13)

⊗ Affichage:

```
int main(){
```

```
    cellule *liste;
```

```
    liste = creation(4);
```

```
    return 0;
```

```
}
```

⊗ Affichage d'une liste:

```
void affichage(cellule *l){
```

```
    while(l != NULL){
```

```
        printf("%d", l->val);
```

```
        l = l->suiv;
```

```
    }
```

Existence d'une val dans une liste chainée

```
int exist(cellule *l, int a){
```

```
    int b = 0;
```

```
    while(l != NULL && b == 0){
```

```
        if(l->val == a){
```

```
            b = 1;
```

```
        else
```

```
            l = l->suiv;
```

```
    return b;
```

(14)

```
int main(){
```

```
    cellule *liste = creation(4);
```

```
    if(exist(liste, 16))
```

```
        printf("existe");
```

```
    else
```

```
        printf("n'existe pas");
```

```
    return 0;
```

```
}
```


① Nombre d'occurrence dans une liste chaînée :

```
int nb_occ(cellule* l, int a) {  
    int c = 0;  
    while (l != NULL) {  
        if (l->val == a) {  
            c++;  
            l = l->suiv;  
        } else {  
            l = l->suiv;  
        }  
    }  
    return c;  
}
```

```
int main() {  
    cellule *liste = Creation();  
    printf("nb occ %d",  
           nb_occ(liste, 18));  
    return 0;  
}
```

② ajout d'un él dans une liste chaînée (à la fin)

```
cellule *ajout_fin (cellule *l, int a) {  
    cellule *p; *tete = l;  
    p = malloc (sizeof (cellule));  
    p->val = a;  
    p->suiv = NULL;  
    while (l->suiv != NULL) {  
        l = l->suiv;  
    }  
    l->suiv = p;  
    return tete;  
}
```