

TP : Liste linéaire chaînée LLC

Développer les algorithmes suivants sur les listes linéaires chaînées :

1. Construire une Llc à partir de n données lues.
2. Longueur d'une Llc.
3. Rechercher dans une Llc l'élément qui a le plus grand nombre d'occurrences.
4. Accès par valeur dans une Llc.
5. Accès par position dans une Llc.
6. Suppression par valeur dans une Llc.
7. Suppression par position dans une Llc.
8. Insertion par position dans une Llc.
9. Interclassement de deux listes ordonnées.
10. Eclater une liste en 2 Llcs selon un critère donné.
11. Trier une Llc par la méthode des bulles.

Quelques bouts de codes

```
struct Maillon
{
    int val ;
    struct Maillon *Suiv ;
};

struct liste {
    struct Maillon *tete;
    struct Maillon *queue;
    int taille;
};
```

```
void cree_llc(struct liste *L) {
    int i,valeur;
    struct Maillon *P,*Q;
    L->tete = NULL ;
    L->queue = NULL ;
    for (i=0;i<L->taille;i++) {
        scanf("%d",&valeur);
        P = Allouer() ;
        Affval(P,valeur);
        Affadr(P,NULL) ;
        if (L->tete != NULL) Affadr(Q,P) ;
        else L->tete = P ;
        Q = P ;
    }
    L->queue = Q ;
}
```

```
void aff_llc(struct liste *L)
{
    struct Maillon *P ;
    int i;

    P = L->tete ;
    for (i=0;i<L->taille-1;i++) printf(" --- ");
    printf(" --- \n");
    while (P != NULL) {
        printf("| %d |--",Valeur(P)) ;
        P = Suivant(P);
    }
    printf("[-\n");
    for (i=0;i<L->taille-1;i++) printf(" --- ");
    printf(" --- \n");
}
```

```
int main ()
{
    struct liste L;
    int occmax, taill, valeur;
    printf("Veuillez entrer les N valeur a chainer : \n");
    scanf("%d",&taill);
    printf("Veuillez entrer les elements a chainer\n");
    L.taille = taill ;
    cree_llc(&L);
    aff_llc(&L);

    ....

    return 0;
}
```