

TP5 : Structures de données en langage C

Parcours MIPC

```
# include <stdio.h>
# define max 50

typedef struct elt_arbre {
    int info;
    struct elt_arbre *sag;
    struct elt_arbre *sad;
} Noeud ;

typedef struct file {
    Noeud *donnee ;
    struct file *suivant ;
} File;

typedef struct {
    File *debut;
    File *fin;
    int taille;
} Var_File;

int Ajouter(Var_File *VF , Noeud *donnee) {
    File *nouvel_element;
    if ((nouvel_element=(File *) malloc (sizeof(File))) ==NULL) return -1;
    nouvel_element->donnee=donnee;
    nouvel_element->suivant = NULL;
    if (VF->taille == 0) VF->debut = nouvel_element;
    else VF->fin->suivant = nouvel_element;
    VF->fin = nouvel_element;
    VF->taille++;
    return 0;
}

Noeud *Retirer(Var_File * VF) {
    File *supp_element;
    Noeud *b;
    int info;
    if (VF->taille == 0) {printf("file vide \n"); return NULL; }
    supp_element = VF->debut;
    b=supp_element->donnee;
```

```

VF->debut = VF->debut->suivant;
if (VF->taille == 1) VF->fin=NULL;
free (supp_element);
VF->taille--;
return b;
}

```

```

void Initialiser(Var_File * VF) {
    VF->taille = 0 ; VF->fin=NULL; VF->debut=NULL;
}

```

```

void parcourir_largeur(Noeud *ab, Noeud *tab[max], int *Nb_element) {
    Noeud *b;
    Var_File * VF;
    Initialiser(VF);
    int i=0;
    if (ab!=NULL) {Ajouter(VF, ab); tab[i]=ab; i++;}
    while (VF !=NULL) {
        b=Retirer(VF);
        if (b->sag!=NULL) {Ajouter(VF, b->sag); tab[i]=b->sag; i++;}
        if (b->sad!=NULL) {Ajouter(VF, b->sad); tab[i]=b->sad; i++;}
    }
    *Nb_element=i;
}

```

```

Noeud *preparer_Noeud(int e) {
    Noeud *n;
    if ((n=(Noeud*)malloc(sizeof(Noeud)))==NULL) return NULL;
    n->info = e;
    n->sag = NULL;
    n->sad = NULL;
    return n;
}

```

```

int ajouter_noeuds(Noeud *b, Noeud *v) {
    Noeud *a,*s;
    a=b; s=b;
    while (a!=NULL) {
        if (v->info < a->info) {s=a; a=a->sag; }
        else if (v->info > a->info) { s=a; a=a->sad; }
        else return -1;
    }
    if (v->info < s->info) s->sag=v;
    if (v->info > s->info) s->sad=v;
    return 0;
}

```

```
void rendre_ab_abo(Noeud *ab, Noeud *aro) {  
    Noeud *tab[max];  
    int Nb_element,i;  
    if (ab==NULL) aro=NULL;  
    parcourir_largeur(ab, tab, &Nb_element);  
    for(i=0; i<=Nb_element-1; i++) ajouter_noeuds(ar, tab[i]);  
}
```

```
void max_min(Noeud *ab) {  
    int maxi, mini;  
    Noeud *a;  
    a=ab;  
    while (a->sag!=NULL) a=a->sag;  
    mini=a->info;  
    printf("min = %d \n",mini);  
    a=ab;  
    while (a->sad!=NULL) a=a->sad;  
    maxi=a->info;  
    printf("max = %d \n",maxi);  
}  
main() {  
}
```