```c
/************************/
/*** Gestion de liste ***/
/************************/
#include "file.h"


/*
 * Creation d'un noeud de valeur x et de suivant suiv
 * retourne le noeud cree
 */
Pliste creerEltListe(PArbre x, Pliste suiv) {
    Pliste ptr;
    if ((ptr = MALLOC(Liste)) == NULL) {
        fprintf(stderr, "ERREUR ALLOCATION MEMOIRE FILE");
        exit(1);
    }
    *ptr = (Liste) { .val = x, .suiv = suiv };
    return ptr;
}

/*
 * Initialisation des tete et queue de la file
 */
void creerFile(SFile *f) {
    f->tete = NULL;
    f->queue = NULL;
}

/*
 * Teste de file vide
 */
bool fileVide(SFile f) {
    if (f.tete == NULL) return true;
    return false;
}

/*
 * Enfilement : on insere l'element
 * tete et queue sont modifiees donc passees par adresse (PrtListe *)
 */
void enfiler(SFile *f, PArbre x) {
    Pliste ptr = creerEltListe(x, NULL);
    if (f->queue) f->queue->suiv = ptr; // queue pointe sur le nouvel element
    else f->tete = ptr;                 // si file vide = nouvel element devient la
    tete
    f->queue = ptr;                     // nouvel element = nouvelle queue
}

/*
 * Defilement : tete et queue sont modifiees donc passees par adresse (PrtListe *)
 */
PArbre defiler(SFile *f) {
    Pliste ptr = f->tete;
    PArbre x;
    if (!ptr) return ARBRENULL;             // File vide
    f->tete = ptr->suiv;                // tete passe au suivant
    if (f->queue == ptr) f->queue = NULL; // File devient NULL tete = queue
    x = ptr->val;                       // on recupere la valeur
    free(ptr);                          // on libere l'element
    return x;
}

```