

```

/*****
3- FONCTIONS DE MANIPULATION D'UNE PILE
*****/
/**
 * Détermine l'élément au sommet d'une pile.
 * Paramètre p : pile à examiner.
 * Paramètre e : élément en sommet de pile retourné.
 * Retourne : 1 si succès et 0 sinon.
 */
int sommet(pile p, int* e)
{
    if ( estPileVide(p) ) return 0;
    *e = p->info;
    return 1;
}

/**
 * Empile un élément sur une pile.
 * Paramètre p : pile sur laquelle empiler.
 * Paramètre e : élément à empiler.
 * Retourne : 1 si succès et 0 sinon.
 */
int empiler(pile* p, int e)
{
    if ( p == 0 ) return 0;
    maillon* m = new maillon;
    m->info = e;
    m->succ = *p;
    *p = m;
    return 1;
}

/**
 * Dépile un élément d'une pile.
 * Paramètre p : pile de laquelle dépiler.
 * Paramètre e : élément dépilé.
 * Retourne : 1 si succès et 0 sinon.
 */
int depiler(pile* p, int* e)
{
    if ( p==NULL || estPileVide(*p) ) return 0;
    *e = (*p)->info;
    maillon* s = (*p);
    *p = (*p)->succ;
    delete s;
    return 1;
}

/*****
4- FONCTIONS DE MANIPULATION D'UNE FILE
*****/

```

```

/**
 * Alloue dynamiquement une file et l'initialise à file vide.
 * Retourne : la file vide créée.
 */
file* creerFile()
{
    file* f = new file;
    f->entree = NULL;
    f->sortie = NULL;
    f->taille = 0;
    return f;
}

/**
 * Initialise une file à "file vide".
 * Usage : pour initialiser une file venant d'être créée par allocation automatique.
 * Paramètre f : file à initialiser.
 */
void initFile(file* f)
{
    if ( f == NULL ) return;
    f->entree = NULL;
    f->sortie = NULL;
    f->taille = 0;
}

/**
 * Détermine la taille d'une file.
 * Paramètre f : file à examiner.
 * Retourne : la taille de f.
 */
int tailleFile(const file *f)
{
    if ( f == NULL ) return -1;
    return f->taille;
}

/**
 * Détermine si une file est vide.
 * Paramètre f : file à tester.
 * Retourne : 1 si f est vide et 0 sinon.
 */
int estFileVide(const file *f)
{
    if ( f == NULL ) return -1;
    return ( f->entree==NULL || f->sortie==NULL );
}

/**
 * Enfile un élément dans une file.
 * Paramètre f : file dans laquelle enfiler.
 * Paramètre e : élément à enfiler.

```

* Retourne : 1 si succès et 0 sinon.

*/

int enfiler(file* f, int e)

```
{
    if ( f == NULL ) return 0;
    maillon* m = new maillon;
    m->info = e;
    m->succ = NULL;
    if ( estFileVide(f) ) f->sortie = m;
    else f->entree->succ = m;
    f->entree = m;
    f->taille += 1;
    return 1;
}
```

/**

* Défile un élément d'une file.

* Paramètre f : file de laquelle défiler.

* Paramètre e : élément défilé.

* Retourne : 1 si succès et 0 sinon.

*/

int defiler(file* f, int* e)

```
{
    if ( f == NULL || estFileVide(f) ) return 0;
    maillon* p;
    p = f->sortie;
    *e = p->info;
    f->sortie = p->succ;
    delete p;
    if ( f->sortie == NULL ) f->entree = NULL;
    f->taille--;
    return 1;
}
```