

## R3.02 Développement efficace (Sae S3.01)

### La structure FIFO

#### Exercice 1

On considère le type file de personnes défini de la manière suivante :

```
typedef struct {
    char nom[50];
    int age;
} tpersonne ;

typedef struct elem{
    tpersonne pers;
    struct elem* svt;
} telement;

typedef struct{
    telement* queue;
    telement* tete;
} tfile;
```

**Question 1/** Définir les procédures pour la création d'une file, le test de la vacuité d'une file, l'obtention de la tête d'une file, l'ajout dans une file (toutes les informations sont des paramètres) et la suppression dans une file.

On considère maintenant le type `tfilePrio` qui définit un ensemble de trois files :

```
typedef struct{
    tfile urgent;
    tfile important;
    tfile standard;
}tfilePrio ;
```

Chaque file a un niveau de priorité. Les personnes à traiter en priorité seront dans la file `urgent`, celles de deuxième priorité seront dans la file `important` et celles sans priorité seront dans la file `standard`.

**Question 2/** Proposez la procédure qui initialise une `tfilePrio` `fp`.

```
void creationPrio(tfilePrio *fp);
```

**Question 3/** Proposez la procédure qui ajoute une `tpersonne p` dans une `tfilePrio fp`. Si la priorité est de 1, l'ajout se fera dans la file urgent, si la priorité est de 2, l'ajout se fera dans la file important et si la priorité est de 3, l'ajout se fera dans la file standard.

```
void ajoutePrio(tfilePrio *fp, tpersonne p, int priorite);
```

**Question 4/** Proposez la procédure qui affiche et enlève d'une `tfilePrio fp` uniquement la prochaine personne à traiter : si possible, on affiche et enlève la tête de file de la file urgent. Dans le cas où cette file est vide, on essaie d'afficher et d'enlever la tête de file de la file important. Si la file important est vide, on enlève et affiche la tête de file de la file standard. Dans le cas où les trois files sont vides, un message est affiché à l'écran.

```
void affichePrio(tfilePrio *fp);
```

-----