

## Programmation : TP 14

**Objectifs** : [OCaml, C] Entrées-sorties, traitement des exceptions.

### [C] Entrées-sorties

On utilisera dans ce td le type et les fonctions disponibles dans `chaine.o` et `chaine.h`. (dans le répertoire `~serveur/PROG/C`) pour manipuler une file de priorité dont chaque élément (une chaîne de caractères) possède un rang (un entier).

Pour compiler on utilisera les commandes suivantes :

```
gcc -W -Wall -c -I ~serveur/PROG/C mon_fichier.c
gcc -W -Wall -o mon_exe ~serveur/PROG/C/chaine.o mon_fichier.o
```

1. Écrire une fonction `void saisie(liste* l)` qui permet l'ajout dans la file d'un élément entré directement sur l'entrée standard (rang et message)<sup>1</sup>. On utilisera la fonction `scanf()` (transparent 220 et faire `man scanf`) et la fonction `insertion()` (voir `chaine.h`),
2. Écrire une fonction `void fprint(liste l)` qui imprime la file dans un fichier dont le nom sera saisi sur l'entrée standard (transparent 214 pour ouverture du fichier, 216 pour l'écriture, 223 pour la fermeture).
3. Écrire une fonction `void supp(liste* l)` qui supprime un élément dont le rang sera saisi sur l'entrée standard.
4. La fonction principale proposera à l'utilisateur l'insertion ou la suppression d'un élément, la sauvegarde dans un fichier de la file ou l'arrêt.

Exemple de trace d'exécution :

```
Saisie = 1, Suppression = 2, Ecriture Fichier = 3, Sortir = 9: 1
entrez le rang suivi du message: 123 Bonjour
```

```
Saisie = 1, Suppression = 2, Ecriture Fichier = 3, Sortir = 9: 1
entrez le rang suivi du message: 12 Bonsoir
```

```
Saisie = 1, Suppression = 2, Ecriture Fichier = 3, Sortir = 9: 3
entrez le nom du fichier: bob.txt
```

```
Saisie = 1, Suppression = 2, Ecriture Fichier = 3, Sortir = 9: 2
entrez le rang de la fiche a supprimer: 124
Suppression d'un element inexistant
```

```
Saisie = 1, Suppression = 2, Ecriture Fichier = 3, Sortir = 9: 9
```

Contenu du fichier `bob.txt` :

```
12      Bonsoir
123     Bonjour
```

### [OCaml] Entrées-sorties, traitement des exceptions

1. Écrire une fonction de lecture du dictionnaire `~serveur/PROG/OCaml/dico.txt` pour fabriquer une liste. Utiliser une boucle infinie (`while true do`) et récupérer l'exception `End_of_file` levée par `input_line` (transparent 215 pour l'ouverture, 219 pour la lecture, 223 pour la fermeture).
2. Écrire une fonction qui sauve les mots de longueur paire dans `dico2.txt` et les mots de longueur impaire dans `dico1.txt`.

<sup>1</sup>On supposera ici que le message fait moins de 20 caractères.