

## Exercice 1

Ecrire un script bash `moyenne.sh` prenant en argument un nombre entier positif  $n$ , et qui affiche tous les nombres de 1 à  $n$ , puis affiche leur moyenne.

## Exercice 2

Ecrire un script bash `monls` prenant en argument un nom de répertoire, et qui parcourt ce répertoire en demandant à l'utilisateur :

- si le parcourt rencontre un fichier, l’affichage de ce fichier,
- si le parcourt rencontre un lien, l’affichage du lien,
- si le parcourt rencontre un sous-répertoire l’affichage du contenu de ce sous-répertoire.

L’interaction avec l’utilisateur se fera au moyen des réponses suivantes :

- o → réaliser l’affichage proposé
- n → refuser l’affichage proposé
- q → terminer l’exécution du script

## Exercice 3

A présent, nous allons utiliser des fonctions. Une fonction permet d’abstraire d’un script, un traitement qui peut être amené à être utilisé dans différents contextes. En plus de factoriser le code d’un script, une fonction permet de le modulariser.

La syntaxe d’une fonction est la suivante :

```
nom_fonction ()  
{  
    < contenu >  
}
```

Une fonction peut être appelée depuis un script ou le shell comme une commande ordinaire :

```
nom_fonction arg1 arg2 ...
```

Dans ce contexte, écrire un script `monrm` permettant de gérer une poubelle (*i.e.*, un répertoire de stockage temporaire de fichiers, avant suppression).

Le principe de ce script est le suivant :

- le script prend en argument une des chaînes suivantes :
  - h affiche un texte d’aide
  - c affiche le contenu de la poubelle
  - e efface un fichier (*i.e.* le met dans la poubelle)
  - v vide la poubelle (avec demande de confirmation)
- chacune des fonctionnalités ci-dessus sera implantée via une fonction spécifique.

## Exercice 4

Enfin, nous allons revenir aux commandes shell, et porter notre attention sur une commande d'édition de fichiers très utile, à savoir `sed`.

Avant de commencer cet exercice, lisez la documentation `sed` jointe, ainsi que la page du manuel de `sed`.

A l'aide de la commande `cat`, créez le fichier `exo1.txt` contenant :

```
Paris Paris
New-York New-York
tokyo Tokyo
londre londre
Le Caire Le Caire
BANGKOK BANGKOK
```

1. Avec la commande `sed`, remplacer les lignes qui contiennent exactement deux mots commençant par une majuscule par 'X'.
2. Remplacer les lignes contenant deux fois le même mot par 'X'.
3. Dans un éditeur, ajouter après 'Tokyo' les lignes "Mexico Mexico" et "Singapour" et enregistrer dans le fichier `exo1b.txt`.
4. Pour chaque ligne de `exo1b.txt` constituée d'exactly deux occurrences d'un même mot, ne garder que la première.
5. Remplacer la troisième ligne par "Rio".