



# **L'épreuve du feu**

# Consignes

- Vous pouvez utiliser le langage de programmation que vous souhaitez.
- Vous devez pouvoir exécuter votre programme depuis le shell.
- En fonction du langage que vous allez utiliser, il se peut que l'appel du programme soit légèrement différent :
  - au lieu d'avoir : `$>./votre_programme arg1 arg2`
  - vous pouvez avoir : `$>ruby votre_programme.rb arg1 arg2`

# Escalier

Créer un script qui affiche un escalier en fonction du premier argument.

Ex. d'appel du script :

```
$> ./escalier 4
```

```
  #  
 ##  
###  
####
```

# Majuscule

Créer un script qui affiche une lettre sur deux en majuscule et le reste en minuscule de la chaîne de caractères du premier argument.

Ex. d'appel du script :

```
$> ./majuscule "Bien le bonjour!"  
bIeN lE bOnJoUr!
```

# Factoriel en récursif

Créer un script qui calcul la factoriel du nombre passé en argument.

Ex. d'appel du script :

```
$> ./factoriel 3  
6
```

# Tri

Créer un script qui affiche les valeurs qui lui sont passées en argument triées par ordre décroissant.

Bonus : implémenter plusieurs algorithmes de tri (tri à bulle, tri sélectif, ... )

Ex. d'appel du script :

```
$> ./tri 4 5 1 3 2  
5 4 3 2 1
```

# Rectangle

Créer un script qui affiche la position du carré passé en premier argument dans le carré du second argument.

Ex. d'appel du script :

```
$> ./rectangle c1.txt c2.txt
```

```
2,1
```

```
$> cat c1.txt
```

```
123
```

```
321
```

```
123
```

```
$> cat c2.txt
```

```
930870
```

```
081235
```

```
973217
```

```
091230
```

```
883700
```



# Anagrammes

Créer un script qui affiche tous les anagrammes du premier argument contenu dans le fichier du second arguments.

Ex. d'appel du script :

```
$> ./anagramme arbre fr.txt  
["arbre", "barre"]
```

```
$> cat fr.txt  
amour  
arbre  
balade  
barre  
beau
```



# Sudoku

Créer un script qui affiche la solution du sudoku passé en premier argument.

Ex. d'appel du script :

```
$> ./sudoku s.txt
```

```
195|784|263
```

```
386|529|147
```

```
472|163|985
```

```
---+---+---
```

```
637|852|419
```

```
859|641|732
```

```
214|397|658
```

```
---+---+---
```

```
923|418|576
```

```
548|976|321
```

```
761|235|894
```

```
$> ./cat s.txt
```

```
195|784|2__
```

```
3__|529|147
```

```
472|1__|985
```

```
---+---+---
```

```
637|852|419
```

```
859|6__|732
```

```
214|397|658
```

```
---+---+---
```

```
92__|418|576
```

```
5__|976|321
```

```
761|235|8__
```

# Factoriel, le retour

Créer un script qui calcul la factoriel du nombre passé en argument, sans exposant.

Ex. d'appel du script :

```
$> ./factoriel 42
```

```
1,405,006,117,752,879,898,543,142,606,244,511,569,936,384,000,000,000
```