

Paradigmes et Langages de Programmation

Haute École d'Ingénierie et de Gestion du Canton de Vaud

5. Haskell / Entrées-sorties

2022

Exercice 1

Écrivez un programme Haskell qui affiche à l'écran une version multilingue de *Hello, world!*. Pour ce faire, utilisez une option de ligne de commande pour indiquer la langue :

- *-en* pour l'anglais
- *-fr* pour le français
- *-it* pour l'italien
- *-es* pour l'espagnol
- *-de* pour l'allemand

Indications :

- Si l'option est omise, le programme affichera un message indiquant son usage.
- Si l'option passée n'est pas supportée, le programme affichera la version anglaise par défaut.

Exercice 2

Écrivez un programme Haskell qui lit un fichier composé de caractères alphanumériques et affiche à l'écran son encodage en code [Morse](#).

Exemple :

```
$ printf 'plp' > tmp
$ ./morse tmp
.-. .-. .-. .-
```

Exercice 3

Écrivez un programme Haskell qui calcule la fréquence de caractères d'une chaîne de caractères passée en ligne de commande et qui écrit dans un fichier un histogramme des fréquences.

Exemple :

```
$ ./freq 109437923954612043294348237
$ cat 109437923954612043294348237
0 * *
1 * *
2 * * * *
3 * * * * *
4 * * * * *
5 *
6 *
7 * *
8 *
9 * * * *
```

Indication : Utilisez group et sort de Data.List pour vous aider à calculer les fréquences de caractères.

Exercice 4

Écrivez un programme Haskell qui permet de copier un fichier d'un emplacement donné à un autre emplacement donné. Autrement dit, votre programme attend deux arguments de ligne de commande lesquels indiquent respectivement la source et la destination.

Exemple :

```
> printf 'Hello , world!\n' > hello.txt
> ghc copy.hs
> ./copy hello.txt salut.txt
> cat salut.txt
Hello , world!
```

Indications :

- Écrivez une première version en utilisant la fonction `copyFile` du module `System.Directory`.
- Écrivez une deuxième version laquelle copie manuellement avec les fonctions `readFile` et `writeFile`.

Exercice 5

Écrivez une fonction Haskell qui permet de récupérer le *nième* élément d'une liste sans utiliser l'opérateur `!!`.

Indications :

- Écrivez une première version qui levera une exception que vous définirez en cas d'erreur.
- Écrivez une deuxième version qui retournera un `Maybe` au lieu de lever une exception.

Bon travail!