

Exercices Haskell

Voici quelques exercices pour mettre en pratique vos connaissances en Haskell. Pour chaque exercice, créez une fonction qui effectue la tâche décrite.

1. Crée une fonction `mySucc` qui prend un entier et retourne cet entier incrémenté de 1.
2. Crée une fonction `myIsNeg` qui prend un entier et retourne un booléen indiquant si l'entier est négatif.
3. Crée une fonction `myAbs` qui prend un entier et retourne sa valeur absolue.
4. Crée une fonction `myMin` qui prend deux entiers et retourne le plus petit des deux.
5. Crée une fonction `myMax` qui prend deux entiers et retourne le plus grand des deux.
6. Crée une fonction `myTuple` qui prend deux éléments de types quelconques et retourne un tuple contenant ces deux éléments.
7. Crée une fonction `myTriple` qui prend trois éléments de types quelconques et retourne un triplet contenant ces trois éléments.
8. Crée une fonction `myFst` qui prend un tuple et retourne le premier élément de ce tuple.
9. Crée une fonction `mySnd` qui prend un tuple et retourne le deuxième élément de ce tuple.
10. Crée une fonction `mySwap` qui prend un tuple et retourne un nouveau tuple avec les éléments

échangés.

11. Crée une fonction ``myHead`` qui prend une liste et retourne le premier élément de la liste. La fonction doit générer une erreur si la liste est vide.

12. Crée une fonction ``myTail`` qui prend une liste et retourne la liste sans le premier élément. La fonction doit générer une erreur si la liste est vide.

13. Crée une fonction ``myLength`` qui prend une liste et retourne la longueur de la liste.

14. Crée une fonction ``myNth`` qui prend une liste et un entier, et retourne l'élément à l'index spécifié par l'entier. La fonction doit générer une erreur si l'index est hors limites ou négatif.

15. Crée une fonction ``myTake`` qui prend un entier et une liste, et retourne une nouvelle liste contenant les premiers éléments jusqu'à l'index spécifié.

16. Crée une fonction ``myDrop`` qui prend un entier et une liste, et retourne une nouvelle liste sans les premiers éléments jusqu'à l'index spécifié.

17. Crée une fonction ``myAppend`` qui prend deux listes et retourne une nouvelle liste contenant les éléments des deux listes.

18. Crée une fonction ``myReverse`` qui prend une liste et retourne la liste inversée.

19. Crée une fonction ``myInit`` qui prend une liste et retourne la liste sans son dernier élément. La fonction doit générer une erreur si la liste est vide.