DÉMO INF-2160 : PARADIGME DE PROGRAMMATION

Démo 2 : Récursion

1. Construire la fonction récursive suivante en Haskell :

$$p(x,0) = 1$$
$$p(x,n) = p\left(x,\frac{n}{2}\right) * p\left(x,\frac{n}{2}\right) \qquad si\ n\ est\ pair$$
$$p(x,n) = p\left(x,\frac{n}{2}\right) * p\left(x,\frac{n}{2}\right) * x \qquad si\ n\ est\ impair$$

   n est une valeur Int, le type de x n'est pas important.
   Essayez votre fonction à l'aide du main suivant :

   ```
   main = print ( p 4 5 )
   ```

   Lorsque compilé, l'exécution du programme devrait afficher 1024.

2. Remplacez le programme principal par l'évaluation suivante :

   ```
   main = print ( [ p 2 x | x <- [ 0 .. 10 ] ] )
   ```

   Ce qui devrait afficher la liste suivante : [1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024].

3. Écrivez une fonction add1s qui reçoit une liste en argument et qui utilisera une compréhension de liste pour ajouter 1 aux éléments de la liste. Par exemple : `main = print ( add1s [1..4] )` devrait afficher [2, 3, 4, 5]
4. Écrivez une fonction diviseur qui reçoit un Int en argument et qui calcule la liste des valeurs qui le divise. Par exemple, pour la valeur 18, nous obtiendrions la liste [1, 2, 3, 6, 9, 18]. Pour cela, utilisez un filtre dans une compréhension de liste.
5. Écrivez une fonction qui reçoit une liste d'élément et retourne l'élément minimum de cette liste.

   ```
   monMinimum :: [Int] -> Int
   ```

   Par exemple, `main = print ( monMinimum [5, 4, 2, 7, 4, 6] )` afficherait 2

6. Écrivez une fonction qui supprime une valeur d'une liste de valeurs en argument.

   supprimer :: Int -> [Int] -> [Int]

   Par exemple, main = print (supprimer 2 [5, 4, 2, 7, 2, 5]) afficherait [5, 4, 7, 2, 5]

7. Utilisez les fonctions monMinimum et supprimer pour construire une fonction de tri sélection.