## Робота №01

*Перші функції*

На основі допоміжного файлу створити файл, в якому надати визначення наступних функцій.

1. Рекурсивна функція *factorial n*, котра вираховує факторіал цілого числа *n*. Передумова - *n≥0.*
2. Функція *listSum xs ys*, котра додає елементи двох списків цілих чисел і повертає список, складений з сум елементів списків *xs ys*. Якщо списки мають різну довжину, то коротший розширюється нулями. Наприклад:
   * listSum [1,2,5] [5, 4,-1] = [6,6,4]
   * listSum [1,2,5] [1] = [2,2,5]
3. Функція *oddEven xs,* котра міняє місцями значення сусідні парних і непарних елементів списку *xs.* Наприклад:
   * oddEven [5,7,6,4,3] = [7,5,4,6,3].
4. Функція *position n xs*, що повертає номер (індекс) першого входження елемента *n* в список цілих чисел *xs*. Індекси елементів починаються з нуля. Якщо елемент не входить в список, то повертається -1. Наприклад:
   * position 5 [3, 5, 91, 0, 5] = 1
   * position 5 [] = -1
5. Функція *set xs*, що формує список, котрий містить всі елементи початкового списку цілих чисел *xs* без дублікатів. Наприклад:
   * set [1, 89, 6, 6, 1] = [1, 89, 6]
6. Функція *union xs ys*, котра обчислює список без дублікатів – об`єднання двох списків цілих чисел *xs* і *ys*.
   * union [1, 2, 1] [3, 2, 2] = [1, 2]
7. Функція *intersection xs ys*, котра обчислює список без дублікатів – перетин двох списків цілих чисел xs і ys.
   * intersection [1, 5, 2, 1, 2] [1, 3, 2, 2] = [1, 2]
8. Функція *factorialsM,* котра будує нескінченний список факторіалів додатних цілих чисел. Можна скористатися функцією *factorial*. Наприклад:
   * take 5 factorialsM = [ 1, 2, 6, 24, 120]

*factorial* :: Integer -> Integer

*listSum*  :: [Int] -> [Int] -> [Int]

*oddEven* :: [Int] -> [Int]

*position*  :: Int -> [Int] -> Int

*set*  :: [Int] -> [Int]

*union, intersection* :: [Int] -> [Int] -> [Int]

*factorialsM* :: [Integer]

Зауваження:

Назва файлу Family01.hs (Family – прізвище студента). Файл включає модуль Family01 і створюється на основі файлу-заготовки HWI01.hs