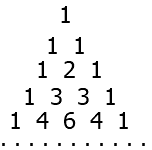
## Робота №02

*Функції*

.

Трикутник Паскаля – нескінченна таблиця біноміальних коефіцієнтів, що має форму трикутника. В цьому трикутнику на вершині та по бокам розташовані одиниці. Кожне інше число дорівнює сумі двох розташованих над ним чисел. Рядки трикутнику симетричні відносно вертикальної осі. На малюнку показано перші п»ять рядків трикутника.

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

1. Функція *sumFl xs*, котра вираховує суму цілих чисел – елементів списку *xs*. Використайте функцію *foldl*.
2. Функція *productFr xs*, котра вираховує добуток цілих чисел – елементів списку *xs*. Використайте функцію *foldr*.
3. Функція *concatFr xs ys*, котра виконує конкатенацію двох списків цілих чисел *xs* і *ys*. Функція повинна мати ту ж семантику, що і оператор (++). Використайте функцію *foldr*. Наприклад:
   * concatFr [5,1,4] [6,7] = [5,1,4, 6,7]
4. Функція *sortInsert xs*, котра сортує список цілих чисел в порядку зростання. Використайте функцію *foldl*. Можна побудувати допоміжну функцію *insert xs v,* що вставляє новий елемент *v* у впорядкований список *xs*. Наприклад:
   * sortInsert [9, 1, 7, 5, 8] = [1,5, 7, 8, 9]
5. Функція *findIndices p xs,* котра знаходить індекси тих елементів списку цілих чисел *xs*, котрі задовольняють предикат *p*. Індекси елементів починаються з нуля.*.* Наприклад:
   * findIndices odd [5,7,6,4,3] = [0,1,4]
6. Функція *allReverse xss,* котра бере список рядків *xss* і перевертає список *xss* і всі рядки, що входять до нього. Використайте функції *map* і *reverse*.Наприклад
   * allReverse [“abc”, “”, “123”, “fg”] = [“gf”,”321”, “”, “cba”]
7. Функція *noDigits xs*, котра вилучає з рядка *xs* всі десяткові цифри. Використайте функції *filte*r і *elem*.
8. Функція *cntGood ps v,* котра вираховує

скільки предикатів зі списку  *ps* задовольняє ціле значення *v.* Наприклад:

* + cntGood [(>0), odd, (<12), (/=5)] 6 = 3

1. Функція *trianglePas* , котра генерує нескінченний список рядків трикутника Паскаля. Використайте функцію *iterate* або *scanl.* Наприклад:
   * take 5 trianglePas = [ [1], [1, 1], [1, 2, 1], [1, 3, 3, 1], [1, 4, 6, 4, 1]]
2. Функція *factorialsM,* котра будує нескінченний список факторіалів. Використати функцію *zipWith*. Наприклад: [1..]
   * take 5 factorialsM = [1, 2, 6, 24, 120]

zipWith (\*) factorialsM

*sumFl, productFr* :: [Integer] -> [Integer]

*concatFr*  :: [Int] -> [Int] -> [Int]

*sortInsert*  :: [Int] -> [Int]

*findIndices*  :: (Int -> Bool) -> [Int] -> [Int]

*allReverse* :: [String] -> [String]

*noDigits*  :: String -> String

*cntGood*  :: [Int -> Bool] -> Int -> Int

*trianglePas* :: [[Integer]]

*factorialM* :: [Integer]

Зауваження:

Назва файлу Family02.hs (Family – прізвище студента). Файл включає модуль Family02 і створюється на основі файла-заготовки HWP02.hs