Lista 7

Efekty obliczeniowe. Programowanie imperatywne

W poniższych zadaniach <u>dopuszczalne jest</u> wykorzystanie funkcji wbudowanych obliczających długość listy, odwracających listę oraz łączących dwie listy, o ile <u>nie wpływają one na drastyczne pogorszenie złożoności obliczeniowej</u>.

Każde zadanie, poza implementacją funkcji, musi posiadać kompletny zestaw testów.

Do wykonania zadań należy wykorzystać mechanizmy poznane na wykładzie nr 6.

1) Trójkąt Pascala jest trójkątną tablicą liczb, w której elementy każdego z wierszy powstają w wyniku sumowania dwóch elementów wiersza poprzedniego:

```
0 1
1 1 1
2 1 2 1
3 1 3 3 1
4 1 4 6 4 1
5 1 5 10 10 5 1
6 1 6 15 20 15 6 1
7 1 7 21 35 35 21 7 1
8 1 8 28 56 70 56 28 8 1
9 1 9 36 84 126 126 84 36 9 1
```

Fig. 1 Trójkqt Pascala, źródło: https://pl.wikipedia.org/wiki/Tr%C3%B3jk%C4%85t_Pascala

Niech zmodyfikowany trójkąt Pascala powstaje na podobnej zasadzie, jednak kolejne wiersze powstają w wyniku sumowania lub odejmowania elementów (naprzemiennie, w zależności od numeru wiersza):

```
0
1
          1
             1
         1
            2
               1
        1 -1
             1 1
      1
        0
           0
         0 -2
  1
      2
         1 -2 -1
```

Napisz funkcje generujące wybrany wiersz zmodyfikowanego trójkąta Pascala:

- a. *modifiedPascalF* wykorzystującą <u>jedynie</u> styl **funkcyjny** (rekurencja, listy, dopasowanie wzorca, funkcjonały, itp.), (OCaml lub Scala) (10 pkt.)
- b. *modifiedPascall* wykorzystującą <u>jedynie</u> styl **imperatywny** (tablice, pętle while, zmienne modyfikowalne, itp.) (OCaml i Scala) (40 pkt.)