

Wrocław, dn. 26.10.2017

Paradygmaty programowania - studia stacjonarne, lista 3, 26.10.2017, godzina 15:15
Czas na rozwiązanie: 70 min.

WAŻNE1: Do każdego zadania przygotuj 3-5 testów sprawdzających poprawność działania!

WAŻNE2: Nie wolno stosować funkcji bibliotecznych i konstrukcji innych niż te przedstawione na wykładzie 3.

1. Napisz funkcję rozdzielającą listę wejściową na dwie podlisty.

W pierwszej podliście mają się znaleźć wszystkie elementy o wartościach parzystych. Dodatkowo wszystkie elementy na pierwszej podliście mają być podniesione do potęgi 2.

W drugiej podliście mają się znaleźć wszystkie elementy o wartościach nieparzystych. Dodatkowo wszystkie elementy na drugiej podliście mają być pomnożone przez 3.

Porządek elementów musi być zachowany. Wynik zwróć w postaci pary list.

Napisz funkcję przy użyciu rekurencji ogonowej i nieogonowej.

Przykład:

wywołanie: `podziel [3;6;8;9;13];;`

wynik : `([36;64],[9;27;39])`

Punkty: 12 (język Ocaml) - 6 za wersję z rekursją ogonową i 6 za wersję z rekursją nieogonową

2. Napisz funkcję zwracającą liczbę liczb nieparzystych. Funkcja ma mieć jeden parametr listę liczb całkowitych.

Napisz funkcję przy użyciu rekurencji ogonowej i nieogonowej.

Przykład:

wywołanie: `liczbanieparzystych [5;4;3;2];;`

wynik : 2

Punkty: 8 (język Scala) - 4 za wersję z rekursją ogonową i 4 za wersję z rekursją nieogonową