

**Paradygmaty programowania - studia stacjonarne, lista 2, 19.10.2017, godzina 17:05**  
**Czas na rozwiązanie: 80 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 2.

1. Napisz funkcję rozdzielającą listę wejściową na trzy podlisty.

W pierwszej podliście mają się znaleźć wszystkie elementy, które spełniają równanie  $a \bmod 10 = 1$ .

W drugiej podliście mają się znaleźć wszystkie elementy, które spełniają równanie  $a \bmod 10 = 2$ .

Na trzeciej podliście mają się znaleźć pozostałe elementy. Dodatkowo do wszystkich elementów na trzeciej podliście należy dodać 5.

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

Przykład:

wywołanie: `podziel[0;11;22;33];;`

wynik : `([11],[22],[5;38])`

Punkty: 8 (język Scala)

2. Napisz funkcję łączącą dwie podane listy liczb całkowitych w pary.

Elementy w liście wyjściowej mają występować w postaci par liczba z listy pierwszej, liczba z listy drugiej. Jeżeli listy różnią się długością uzupełniamy brakującą liczbą 0.

Zwróć szczególną uwagę na optymalność złożoności obliczeniowej i pamięciowej.

Przykład:

wywołanie: `polacz [5;4;3;2] [1;2;3;4;5;6];;`

wynik : `[(5,1);(4,2);(3,3);(2,4);(0,5);(0,6)]`

Punkty: 6 (język Scala)

3. Napisz funkcję *zamien* dzielącą daną listę na dwie podlisty w zadanym punkcie i zwracającą listę powstałą z zamiany tych podlist miejscami.

Przykład:

wywołanie: `zamien [1;7;5;6] 2;`

Wynik: `[5;6;1;7]`

Punkty: 7 (język Ocaml)