

Programowanie funkcyjne — kolokwium nr 2, 28.01.2021

Instrukcja: Każde zadanie należy przesłać na Pegaza w oddzielnym pliku: zadanie1.hs, zadanie2.hs i zadanie3.erl. Plików nie należy zipować. Rozwiązania muszą się poprawnie kompilować. W rozwiązaniach nie można korzystać z modułów innych niż standardowe; niedozwolone jest użycie polecenia `import`. Zadania 1 i 2 napisać w Haskellu, zadanie 3 — w Erlangu. Rozwiązania nie spełniające powyższych wymogów nie będą oceniane. Punktacja: 10 punktów za każde zadanie.

Zadanie 1. Napisać bezpunktowo funkcję $f :: [Int] \rightarrow [Int]$, określoną poniżej. Odpowiedź należy uzasadnić, na przykład prezentując kolejne kroki dochodzenia do rozwiązania.

```
f list = map (\(a,b) -> 3 * (a+b)) (zip list [1..10])
```

Zadanie 2. Napisać funkcję `usun :: String -> String -> String`, która ze stringu podanego jako pierwszy argument usuwa wszystkie znaki, które występują w stringu podanym jako drugi argument. Definicja funkcji musi rozpoczynać się od wywołania `foldl` lub `foldr`. Przykład:

```
usun "Ala ma kota" "kra" = "Al m ot"
```

Zadanie 3. Napisać moduł, który uruchomi n procesów ($n \geq 3$). Każdy z procesów ma: (a) odbierać listę liczb, (b) wypisywać ją na konsolę, (c) dołączać na początku listy sumę dwóch pierwszych elementów, (d) skracać listę do dwóch pierwszych elementów, jeżeli jej długość przekracza 4 i (e) przekazywać listę następnemu procesowi w cyklu, w nieskończonej pętli. Moduł ma udostępniać funkcję `start(n, k)`, która uruchomi procesy w stosownym cyklu tak, by całość wypisywała kolejne listy, poczynając od $[k, 1]$.

Przykładowo, wywołanie `start(n, 1)`, z dowolnym $n \geq 3$, powinno spowodować wypisanie następujących list:

```
[1, 1]
[2, 1, 1]
[3, 2, 1, 1]
[5, 3]
[8, 5, 3]
[13, 8, 5, 3]
[21, 13]
...
```