

Programowanie funkcyjne — kolokwium nr 1, 27.11.2019

Instrukcja: Rozwiązania należy przesłać do godziny 9:45, w jednym mailu, na adres kolokwium.pf@gmail.com. Każde zadanie należy przesłać w oddzielnym pliku: zadanie1.hs, zadanie2.hs i zadanie3.hs. Plików nie należy zipować. Rozwiązania muszą się poprawnie kompilować. W rozwiązaniach nie można korzystać z modułów innych niż standardowe; niedozwolone jest użycie polecenia import. Rozwiązania nie spełniające powyższych wymogów nie będą oceniane. Punktacja: 10 punktów za każde zadanie. Uwaga: korzystanie z internetu poza wyznaczonym czasem skutkuje automatycznym otrzymaniem 0 punktów.

Zadanie 1. Dla liczby naturalnej w każdym kroku tworzymy liczbę będącą sumą jej cyfr (w zapisie dziesiętnym) i postępujemy tak, aż dojdziemy do liczby jednocyfrowej. Napisać funkcję `sevens :: Int → [Int]`, zwracającą n pierwszych liczb, dla których powyższy proces skończy się na liczbie 7. Na przykład `sevens 4 = [7, 16, 25, 34]`. Można też zauważyć, że do `sevens 1000` należy np. liczba 8881, bo $8881 \rightarrow 25 \rightarrow 7$.

Zadanie 2. Palindrom nazywamy *zbalansowanym*, jeżeli składa się on wyłącznie z liter a i b , i liczba liter a jest równa liczbie liter b . Napisać funkcję `bp :: Int → [String]`, która dla danego n zwróci listę zbalansowanych palindromów o długości n .

Zadanie 3. *Miłorzęb* to struktura danych o funkcjonalności przypominającej listę, która umożliwia: dokładanie elementów na początek i na koniec w czasie stałym, odczytanie elementów po kolei w czasie liniowym względem liczby elementów oraz usunięcie wszystkich elementów dołożonych na początek („urwanie lewego listka”) i na koniec („urwanie prawego listka”) w czasie stałym (nie liczymy czasu potrzebnego na zwolnienie pamięci). Stworzyć typ `Mb a`, przechowujący elementy typu a w miłorzębie, i zdefiniować funkcje:

```
dnf :: Mb a -> a -> Mb a
dnk :: Mb a -> a -> Mb a
mb2list :: Mb a -> [a]
ull :: Mb a -> Mb a
upl :: Mb a -> Mb a
```

Funkcje mają, odpowiednio, wstawiać element na początek i na koniec miłorzębu, zamieniać miłorzęb na listę (z elementami w odpowiedniej kolejności, tzn. od początku miłorzębu do końca) oraz urywać lewy i prawy listek. Złożoność funkcji powinna być taka, jak opisano wcześniej.