Programowanie funkcyjne — kolokwium nr 1, 7.12.2022

Instrukcja: Każde zadanie należy przesłać na Pegaza 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. Napisać funkcję o sygnaturze

`cztery :: [Int] -> Int`,

która dla podanej listy liczb całkowitych zwróci długość najdłuższego spójnego podciągu o sumie elementów podzielnej przez 4.

Zadanie 2. Napisać funkcję o sygnaturze

`wyniki :: Double -> [(String,Double)] -> [(String,String)]`,

która w pierwszym argumencie otrzymuje maksymalną liczbę punktów do uzyskania z testu, w drugim zaś otrzymuje listę par, które na pierwszej współrzędnej przechowują dane osób piszących test, a na drugiej liczbę uzyskanych punktów. Funkcja ta ma zwracać listę par zmodyfikowanych w następujący sposób:

- Jeśli na pierwszej współrzędnej na początku lub na końcu napisu znajdują się spacje, należy je usunąć (przed pierwszym znakiem, który nie jest spacją, **oraz** za ostatnim znakiem, który nie jest spacją; można założyć, że napisy nie składają się tylko ze spacji). Poza tym napis ma pozostać bez zmian.
- Wynik punktowy z drugiej współrzędnej ma zostać zmieniony w zależności od tego, w jaki przedział procentowy maksymalnego wyniku wpada:

  $(90\%, 100\%] \rightarrow$ `"5.0"`
  $(80\%, 90\%] \rightarrow$ `"4.5"`
  $(70\%, 80\%] \rightarrow$ `"4.0"`
  $(60\%, 70\%] \rightarrow$ `"3.5"`
  $(50\%, 60\%] \rightarrow$ `"3.0"`
  $[0\%, 50\%] \rightarrow$ `"2.0"`

  Jeśli wynik punktowy jest mniejszy od zera lub większy od maksymalnej liczby punktów, należy go zamienić na napis `"Nieprawidłowe dane"`.

Przykładowo, wywołanie

`wyniki 100.0 [(" Arnold ",60.4), ("Franek",75), (" Tim", 991.93)]`

powinno zwrócić

`[("Arnold","3.5"),("Franek","4.0"),("Tim","Nieprawidłowe dane")]`.

Zadanie 3. *Bluszcz skierowany* to struktura danych, która pozwala na: dołączenie elementu (`de`), odczytanie elementu ostatnio dołączonego (`oe`), usunięcie elementu ostatnio dołączonego (`ue`), podanie liczby elementów równych elementowi dołączonemu jako pierwszy (`le`) oraz zamianę całej struktury na listę (`bsk2l`), przy czym dowolne dwa elementy dołączone w następujących po sobie operacjach `de` muszą być sąsiadami na liście. Zamiana na listę powinna być wykonalna w czasie liniowym względem łącznej liczby elementów, natomiast pozostałe operacje — w czasie stałym. Zdefiniować typ `Bsk a`, służący do przechowywania elementów typu `a` w bluszczu skierowanym, oraz następujące funkcje, realizujące opisane wyżej operacje z odpowiednią złożonością:

```
de :: Bsk a -> a -> Bsk a
oe :: Bsk a -> a
ue :: Bsk a -> Bsk a
le :: Eq a => Bsk a -> Integer
bsk2l :: Bsk a -> [a]
```

Funkcje `oe`, `ue` i `le` nie są zdefiniowane dla bluszczu pustego.