

Programowanie funkcyjne — kolokwium nr 2, 22.01.2025

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; nie-dozwolone 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. Uwaga: korzystanie z internetu poza wyznaczonym czasem skutkuje automatycznym otrzymaniem 0 punktów.

Zadanie 1. Napisać program, który wczyta ze standardowego wejścia serię liczb naturalnych dodatnich zakończoną zerem (każda liczba w osobnym wierszu) i wypisze na standardowe wyjście ich największy wspólny dzielnik. W rozwiązaniu należy w istotny sposób użyć funkcji `foldl/foldr`. Przykładowo, po wpisaniu liczb 18, 24, 15 i 0, program powinien wypisać liczbę 3. Uwaga: liczby na wejściu należy samodzielnie parsować znak po znaku, wczytując je za pomocą `getChar` lub `getLine`.

Zadanie 2. Napisać bezpunktowo funkcję

```
nmcp :: Integer -> Integer,
```

znajdującą najmniejszy czynnik pierwszy podanej liczby, równej co najmniej 2. Można założyć, że dostępna jest bezargumentowa funkcja `lpierwsze`, zwracająca nieskończoną, uporządkowaną rosnąco listę liczb pierwszych. Można definiować funkcje pomocnicze, o ile są również napisane bezpunktowo.

Zadanie 3. Napisać moduł uruchamiający N procesów, które roboczo numerujemy od 1 do N , przy czym $N \geq 3$. Proces o numerze i odbiera (komunikat zawierający) liczbę x , wypisuje na ekran parę (i, x) i przesyła *następnemu* procesowi *zmodyfikowaną* liczbę x . *Zmodyfikowana* liczba to $2x$, jeśli proces ma numer parzysty, oraz $x-1$, jeśli proces ma numer nieparzysty. Przez *następny* proces rozumiemy proces o numerze $(i \bmod N) + 1$. Komunikaty przesyłane pomiędzy procesami nie mogą zawierać nic oprócz liczby x . Z wyjątkiem obowiązkowych parametrów standardowych poleceń, w programie nie wolno przekazywać list (ani w komunikatach, ani w parametrach).

Moduł ma udostępniać funkcję `start(N, X)`, która uruchomi procesy w stosownym cyklu tak, by całość wypisywała wyniki działania procesów, poczynając od procesu o numerze 1, któremu zostanie przekazana początkowo liczba X . Przykładowo, wywołanie `start(5, 2)` powinno spowodować wypisanie następujących par:

$(1, 2), (2, 1), (3, 2), (4, 1), (5, 2), (1, 1), (2, 0), (3, 0), (4, -1), (5, -2), \dots$