

# PRACOWNIA Z KURSU JEZYKA ERLANG

## LISTA 2

Można i należy używać modułów `lists` oraz `string`. Proszę używać odpowiedzialnie operatora `++`. Wszystkie zadania na bieżącej liście są warte 1 punkt.

1. Napisz funkcje znajdujące zadane zbiory liczb. Pierwsza funkcja `zaprzyjaznione/1` zwraca listę par liczb zaprzyjaźnionych<sup>1</sup> nie większych niż  $n$ . Druga funkcja `rozklad/1` oblicza rozkład liczby  $n$  na czynniki pierwsze<sup>2</sup> i zwraca jako wynik listę par  $[(p_1, w_1), (p_2, w_2), \dots, (p_k, w_k)]$  taką, że  $n = p_1^{w_1} * p_2^{w_2} * \dots * p_k^{w_k}$  oraz  $p_1, \dots, p_k$  są różnymi liczbami pierwszymi.

```
1> zaprzyjaznione(1300).  
[{220,284},{1184,1210}]  
2> rozklad(756).  
[{2,2},{3,3},{7,1}]
```

2. Napisz program, który będzie czytał tekst ze standardowego wejścia, formatował go i drukował na standardowe wyjście. Tekst wejściowy jest zorganizowany w akapity, które są oddzielone co najmniej jedną pustą linią lub linią składającą się z wyłącznie z białych znaków. Każda wiersz wejściowy może się zaczynać i kończyć jakąś ilością białych znaków poza znakiem końca linii.

Program ma przyjmować jako parametry dwie liczby. Pierwsza to szerokość linii. Drugi jest opcjonalny i oznacza ilość spacji użytych do wcięcia pierwszego wiersza akapitu.

Akapity w wyjściowym tekście mają być oddzielone dokładnie jedną pustą linią. Pierwszy wiersz akapitu ma być wcięty zgodnie z parametrem wejściowym programu. Linia ma być wyrównana do lewej i do prawej, tzn. pierwsza i ostatnia kolumna ma zawierać odpowiednio początek pierwszego słowa i koniec ostatniego słowa. W przypadku ostatniej linii akapitu tekst musi być wyrównany do lewej.

3. Zaimplementuj drzewa samo-balansujące się (*ang. splay tree*<sup>3</sup>). Każdy wierzchołek drzewa powinien składać parę `{Klucz,Wartość}`. Napisz następujące operacje:

**new/1** Tworzy nowe drzewo splay i zwraca krotkę `{ok,Drzewo}`.

**insert/3** Do drzewa  $T$  wstawia węzeł o kluczu  $K$  i wartości  $V$ . Zwraca `{ok,new}`. W przypadku, kiedy węzeł o podanym kluczu już istnieje, stara wartość zostaje zastąpiona, a wynikiem funkcji jest `{ok,StaraWartość}`.

**delete/3** Usuwa z drzewa  $T$  węzeł o kluczu  $K$  i zwraca parę `{ok,Wartość}`.

**search/2** Szuka w drzewie  $T$  węzła o kluczu  $K$  i zwraca parę `{ok,Wartość}`.

**split/2** Dzieli drzewo  $T$  na dwa poddrzewa względem wierzchołka o kluczu  $K$  i zwraca parę `{ok,{DrzewoLewe,DrzewoPrawe}}`.

**join/2** Łączy dwa drzewa  $T_1$  i  $T_2$  w jedno i zwraca `{ok,NoweDrzewo}`.

Jeśli klucza nie znaleziono, funkcje zwracają `{error,not_found}`.

Lista i materiały znajdują się pod adresem

<http://cahirwpz.cs.uni.wroc.pl/main-pl/erlang-language-summer-2010/>

Krystian Baclawski

---

<sup>1</sup>[http://pl.wikipedia.org/wiki/Liczby\\_zaprzyjaznione](http://pl.wikipedia.org/wiki/Liczby_zaprzyjaznione)

<sup>2</sup>[http://pl.wikipedia.org/wiki/Rozk\OT4\lad\\_na\\_czynniki](http://pl.wikipedia.org/wiki/Rozk\OT4\lad_na_czynniki)

<sup>3</sup>[http://pl.wikipedia.org/wiki/Drzewo\\_splay](http://pl.wikipedia.org/wiki/Drzewo_splay)