

Zadanie 6.1

Zapisz w pliku program ze slajdu 70. Skompiluj i przetestuj jego działanie dla wszystkich 3 wersji kodu.

Zadanie 6.2

Skompiluj poniższy program do pliku wykonywalnego. Przeanalizuj kod. Przetestuj działanie programu.

```
module Main where

import System.Environment

main = do
    [f,g] <- getArgs
    s <- readFile f
    writeFile g s
```

Wskazówka: Funkcja **getArgs** pobiera argumenty z linii poleceń.

Zadanie 6.3

Napisz program **html2txt** konwertujący plik HTML na plik tekstowy. W ramach konwersji mają zostać usunięte wszystkie znaczniki HTML, tzn. dowolny tekst między znakami < i >. Program ma otrzymywać dwa parametry z linii poleceń: nazwę pliku do konwersji i nazwę pliku wynikowego.

Wskazówka: Do usuwania znaczników HTML należy zdefiniować funkcję **clear** typu:

```
clear :: String -> Bool -> String
```

Argument typu logicznego określa, czy znajdujemy się aktualnie (w trakcie przetwarzania tekstu) wewnątrz znacznika HTML (wartość True), czy nie.

Zadanie 6.4

(*) Napisz program **sortfile** sortujący plik tekstowy. Program ma otrzymywać dwa parametry z linii poleceń: nazwę pliku do sortowania i nazwę pliku wynikowego.

Wskazówka: Skorzystaj z przedstawionych na wykładzie funkcji: **mergeSort**, **lines** i **unlines**.

Zadanie 6.5

(*) Napisz program **searchFile**, który przeszukuje plik tekstowy i zwraca numery linii, w których znalazł szukane słowo. Program ma otrzymywać dwa parametry z linii poleceń: szukane słowo i nazwę pliku.

Zadanie 6.6

Dany jest plik tekstowy [pesel.txt](#) zawierający numery PESEL z XX w. Napisz program, który:

1. wyznacza liczbę numerów PESEL kobiet (przedostatnia liczba w kodzie jest parzysta);
2. (*) wyznacza liczbę osób urodzonych w grudniu;
3. wyznacza liczbę osób urodzonych w poszczególnych dekadach (wynik jest listą 10-elementową).