

Paradygmaty programowania

Lista 3

Zad. 1. [4p] Napisać w OCaml funkcję o typie $'a\ list * int \rightarrow 'a\ list$ dokonującą podziału zadanej listy na dwie podlisty: listę elementów przed (włącznie) i listę elementów po elemencie o zadanej pozycji, zwracającą listę powstałą z połączenia podlist w odwróconej kolejności.

Zad. 2. [4+2p] Wykorzystując mechanizm dopasowania wzorca napisać w OCaml funkcję typu $'a\ list \rightarrow 'a\ list$ (jedną!) stosującą rekurencję ogonową, która przekształca wejściową listę do postaci, w której każdy element listy jest powtórzony liczbę razy równą pozycji elementu w liście wejściowej. Zadanie wykonać następnie w Scala.