

Zadanie 1.

Napisać polimorficzne funkcje `remove` oraz `qsort` takie, że

- ▶ `remove element list` usuwa z listy `list` wszystkie wystąpienia elementu `element`.
Na przykład `remove 'a' "ała ma kota" = "l m kot"`
- ▶ Funkcja `qsort` powinna sortować listę za pomocą algorytmu *Quick Sort*.

Zadanie 2.

W pliku `personalinfo.hs` zdefiniowany jest typ `PersonalInfo`, który przechowuje imię, nazwisko oraz rok urodzenia pracownika. Zadanie polega na zdefiniowaniu funkcji `eqFunction` oraz `showFunction` tak, aby umożliwiły one uczynienie tego typu instancją klas `Eq` oraz `Show`.

Element tego typu powinny być wyświetlane linijka po linijce. A więc na przykład `PI "Jan" "Kowalski" 1980` powinien zostać wyświetlony tak:

```
Name: Jan
```

```
Surname: Kowalski
```

```
Birthyear: 1980
```

Zadanie 3.

Korzystając z powyższej interpretacji funkcji `Poprzednik` i `Nastepnik` w pliku `calkowite.hs` napisać funkcję `toInt :: Calkowite -> Int`, która będzie przekształcała elementy typu `Calkowite` na "zwykłe" liczby całkowite. Napisać także funkcję `intToCalkowite` działającą w odwrotny sposób.

Zadanie 4.

W pliku `tree.hs` uzupełnić definicje funkcji `isEmpty`, `size` oraz `depth`, które zwracają odpowiednio informację o tym, czy drzewo jest puste, jego rozmiar oraz głębokość.