

Praktikum 5

Listid

Esitavad järjestatud elementide korteeže

```
[a, d, f, [s, f, [],d]]
```

List unifitseerub:

- ühe muutujaga

```
List = [a, d, f, [s, f, [],d]]
```

- Listi erinevaid osi adresseerivate muutujatega, kui on mitte-tühi list

```
[Head|Tail]
```

kus

Head (listi pea) - listi ilmutatult viidatavad esimesed elemendid

Tail (listi saba) - ülejäänud listi elemendid

| - eraldussümbol.

Näiteid :

```
[H|T], [_|T], [H|_], [El1, El2,El3|Tail] jne .
```

Näited

% Elemendi sisalduvuse kontroll (pearekursioon - sügavuti rekursioon)

```
my_member(_, []) :- fail.
```

```
% Tühja listi puhul meid El ei huvita.
```

```
my_member(El, [El|_]).
```

```
% El on esimene, ülejäänud meid ei huvita.
```

```
my_member(El, [_|Tail]) :- my_member(El,Tail).
```

```
% El ei ole esimene. Kuna esimene element ei
```

```
% ole El, siis see element meid ei huvita.
```

```
% Viskame selle elemendi listist välja ja otsime
```

```
% elementi uuesti ühe võrra väiksemast listist.
```

```
?- my_member(2, [3,4,2,5]).
```

```
true .
```

Ülesanne

1. Kirjutada **rekursiivne** reegel `viimane_element/2`, mis leiab listi viimase elemendi.

```
?- viimane_element(X, [a, x, d, c]).  
X = c.
```

NB! Listi ümber keerata ei tohi.

2. Kirjutada reegel `on_palindroom/1`, mis kontrollib, kas etteantud list on palindroom.

```
?- on_palindroom([a, a, b, b, a, a]).  
true.
```

```
?- on_palindroom([a, a, b, c, a, a]).  
false.
```

3. Kirjutada reegel `paki/2`, mis elimineerib listist järgnevad korduvad elemendid.

```
?- paki([a, b, b, b, a, a, c, b, b], X).  
X = [a, b, a, c, b].
```

4. Kirjutada reegel `duplitseeri/2`, mis kahekordistab elemendid etteantud listis.

```
?- duplitseeri([a, b, b, c], X).  
X = [a, a, b, b, b, b, c, c].
```

5. Kirjutada reegel `kordista/3`, mis kordistab listi kõiki elemente etteantud arv korda.

```
?- kordista([a, b, b, c], 3, X).  
X = [a, a, a, b, b, b, b, b, b, b, c, c, c].
```