

## Praktikum PR03

### 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|_],
[E1, E2,E3|Tail] jne .
```

### Näited

% Elemendi sisalduvuse kontroll (pearekursioon - sügavuti rekursioon)

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

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

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

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

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

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

```
% ole EI, 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 `suurim/2`, mis kontrollib etteantud listist järjest paarikaupa elemente ja paneb väljundlisti elemendi, mis on antud paari elementidest suurim. Kui võrreldakse elementi ja tühilisti, siis väljundlisti tuleb panna element.

```
?- suurim([1,2,6,8,5], X).  
X = [2,6,8,8,5].
```

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

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

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

```
?- duplikeeri([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, c, c, c].
```

6. Kirjutada reegel `vordle_predikaadiga/3`, mis võrdleb etteantud predikaadiga listi kõiki liikmeid ja paneb väljundlisti need elemendid, mis vastavad tingimustele. Võrdluspredikaadid on:

- `paaritu_arv`
- `paaris_arv`
- `suurem_kui(X)`

Võrdluspredikaadid tuleb ise implementeerida.

```
?- vordle_predikaadiga([4,5,6,7], [paaritu_arv], X).  
X = [5, 7].  
?- vordle_predikaadiga([4,5,6,7], [suurem_kui, 5], X).  
X = [6, 7].
```