

4. laboratorijas darbs

1. Uzrakstīt predikātu `elem(list, integer, integer)`. Rezultāts: saraksta elementa *vērtības* iegūšana, norādot elementa indeksu. Pirmā elementa indekss ir **1**.

```
elem([4, 5, 6], 2, X)      % X=5
elem([4, 5, 6], 5, X)      % No Solution
```

2. Uzrakstīt predikātu `pos1(list, integer, integer)`. Rezultāts: saraksta elementa *indeksa* iegūšana, norādot elementa vērtību. Ja sarakstā ir vairāki elementi, tiks iegūts *pirmā* elementa indekss. Pirmā elementa indekss ir **1**.

```
pos1([4, 5, 6], 5, X)      % X=2
pos1([4, 5, 6], 1, X)      % No Solution
```

3. Uzrakstīt predikātu `pos2(list, integer, integer)`. Rezultāts: saraksta elementa *indeksa* iegūšana, norādot elementa vērtību. Ja sarakstā ir vairāki elementi, tiks iegūts *pirmā* elementa indekss. Saraksta apstrāde notiek virzienā „*uz priekšu*”. Var izmantot papildus predikātus. Pirmā elementa indekss ir **1**.

```
pos2([4, 5, 6], 5, X)      % X=2
pos2([4, 5, 6], 1, X)      % No Solution
```

4. Uzrakstīt predikātu `max(list, integer)`. Rezultāts: *maksimālā* elementa iegūšana. Saraksta apstrāde notiek virzienā „*uz priekšu*” (pirmais saraksta elements ir *sākotnējais* maksimālais elements). Var izmantot papildus predikātus.

```
max([4, 50, 6], X)         % X=50
max([4, 50, 60], X)        % X=60
```

5. Uzrakstīt predikātu `max_pos(list, integer)`. Rezultāts: *maksimālā pozitīvā* elementa iegūšana. Ja sarakstā nav neviena pozitīva elementa, rezultāts ir **-1**. Piezīme: elements **0** nav pozitīvais vai negatīvais elements.

```
max_pos([2, 3, -7], X)     % X=3
max_pos([-2, -3, -7], X)   % X=-1
```

6. Uzrakstīt predikātu `min_pos(list, integer)`. Rezultāts: *minimālā pozitīvā* elementa iegūšana. Ja sarakstā nav neviena pozitīva elementa, rezultāts ir **-1**. Piezīme: elements **0** nav pozitīvais vai negatīvais elements.

```
min_pos([-2, 3, 7], X)     % X=3
min_pos([-2, -3, -7], X)   % X=-1
```