

# Domácí úkol 11

**Příklad 11.1** Naprogramujte predikát `maxmax/2`, který dostane v prvním argumentu seznam seznamů celých čísel a do druhého argumentu unifikuje maximální prvek těchto seznamů (najde globální maximum). Predikát naprogramujte tak, aby fungoval v módu

```
maxmax( +ListOfLists, ?GlobalMax )
```

Můžete předpokládat, že první argument bude správně utvořený seznam seznamů celých čísel, bude obsahovat alespoň jeden seznam a každý jeho podseznam bude obsahovat alespoň jedno číslo. Predikát musí na zadaném seznamu uspět právě jednou (i když by se tam maximální prvek vyskytoval vícekrát). Navíc nesmí nabídnout při dotazování v interpretru možnost doptat se na další výsledek (tedy interpreter odpoví a ihned přejde do módu, kdy čeká na další dotaz).

V řešení si můžete definovat další pomocné predikáty. Nesmíte načítat žádné knihovny ani používat vestavěné predikáty (`member/2`, `append/3`, ...) bez toho, abyste uvedli jejich implementaci.

Následuje několik příkladů použití:

```
?- maxmax( [[1,3,2],[2]], X ).  
X = 3.
```

```
?- maxmax( [[3],[3,2]], X ).  
X = 3.
```

```
?- maxmax( [[1,3,2],[2]], 3 ).  
true.
```

```
?- maxmax( [[1,3,2],[5]], 3 ).  
false.
```

Následně nakreslete SLD-strom výpočtu pro vaši implementaci pro níže uvedený dotaz. Pokud jsou ve výpočetním stromě místa, kde došlo k ořezání/upnutí, naznačte je čarou k odstraněné větvi a uveďte, zda se jedná o ořezání, nebo upnutí.

```
?- maxmax( [[1,3],[3]], X ).
```