

## Domácí úkol číslo 1: Rekurze

- 1) Naprogramujte predikát **trojuhela/1**, jeho vstupem je kladné celé číslo, udávající počet řádek trojúhelníku. Výstupem je obrázek trojúhelníku, viz. následující příklad. Pro lepší nakreslení doplňte čísla mezerou nebo nulou tak, aby byla vizuálně všechna stejně dlouhá.

?- trojuhela(5).

```
  01
 02 03
04 05 06
07 08 09 10
11 12 13 14 15
```

- 2) Naprogramujte predikát **trojuhelaB/1**, jeho vstupem je kladné celé číslo. Výstupem je obrázek trojúhelníku, viz. následující příklad.

?- trojuhelaB(4).

```
1
121
12321
1234321
123333321
1222222221
1111111111111
```

- 3) Naprogramujte predikát **diamant/1**, jeho vstupem je kladné celé číslo  $N$ . Výstupem je obrázek diamantu, viz. následující příklad. Pro nakreslení budete potřebovat  $N^2$  čísel, pro lepší nakreslení doplňte čísla mezerou nebo nulou tak, aby byla vizuálně všechna stejně dlouhá.

?- diamant(3).

```
  1
  4 2
 7 5 3
 8 6
 9
```

?- diamant(4).

```
  01
  05 02
 09 06 03
13 10 07 04
14 11 08
 15 12
 16
```

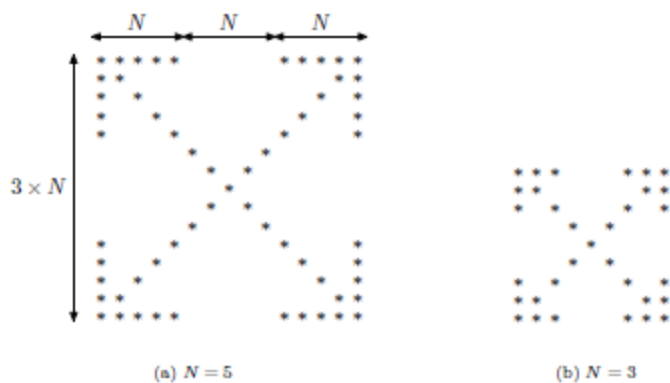
- 4) Naprogramujte predikát **kriz/1**, jeho vstupem je kladné celé číslo. Výstupem je obrázek kříže, viz. následující příklady. Pro lepší nakreslení doplňte čísla mezerou nebo nulou tak, aby byla vizuálně všechna stejně dlouhá.

?- kriz(3).	?-kriz(6).
1 2	1        2
5	5     6
4 3	9 10
	12 11
	8     7
	3       4

- 5) Naprogramujte predikát **stopka/1**, vstupem je kladné celé číslo. Výstupem je obrázek značky stop, viz. následující příklady.

?- stopka(3).	?-stopka(5).
RRR	RRRRR
R  R	R     R
R    R	R     R
R WWW R	R     R
R    R	R     R
R  R	R     R
RRR	R WWWWW R
	R     R
	R     R
	R     R
	RRRRR

- 6) Naprogramujte predikát **sipky/1**, jehož vstupem je kladné celé číslo. Výstupem je obrázek šipek, viz. následující příklady.



**7) Úkol: "Šílená aritmetika" (Rozhodně takhle s čísly v Prologu normálně nepracujete!)**

Vaším úkolem je napsat predikáty pro aritmetické operace, přičemž nesmíte používat vestavěný operátor "is".

Číslo budou reprezentována následovně: 0 ... 0, 1 ... succ(0), 2 ... succ(succ(0)), 3 ... succ(succ(succ(0))) atd.

Definici čísla už máte připravenou zde (predikát "cislo"). Vyzkoušejte si v ECLiPSe zavolat dotaz cislo(N) a několikrát klikněte na "more".

**cislo(0).**

**cislo(succ(N)) :- cislo(N).**

Pro účely testování můžete používat následující "predikát", který vám převede číslo do "succ"-reprezentace. (nikde jinde se ale operátor "is" už ve Vašem kódu vyskytnout nesmí).

**natural2succ(0,0).**

**natural2succ(N,succ(S)) :- N > 0, M is N-1, natural2succ(M,S).**

- A. Napište predikát "soucet(A,B,Soucet)" takový, že bude platit  $A+B = \text{Soucet}$ . Například pro dotaz `?-soucet(succ(succ(0)), succ(0), X)` byste měli dostat  $X = \text{succ(succ(succ(0)))}$  a podobně pro dotaz `?-soucet(succ(succ(0)), X, succ(succ(succ(0))))` byste měli dostat  $X = \text{succ(0)}$  atd.
- B. Pomocí predikátu součet napište analogický predikát "rozdil(A,B,Rozdil)".
- C. S využitím predikátu pro součet napište predikát pro součin dvou čísel.
- D. Dále si naprogramujte predikát "mensiNez(A,B)".