

```
cifra(0).
cifra(1).
cifra(2).
cifra(3).
cifra(4).
cifra(5).
cifra(6).
cifra(7).
cifra(8).
cifra(9).
```

```
% v komentáři za pomlčkou následuje co vepisovat do konzole - ukázky
% příkazy v konzoli nutno ukončovat tečkou! Jinak si prolog myslí, že číslo neskončilo
% chci příkazy v konzoli ukončit - quit. a program znovu přelož - Run as/Interpreted Single File
```

```
% jednociferná čísla - císlo(5). - císlo(X) - ;
císlo(X) :- cifra(X).
```

```
% dvojciferná čísla - císlo(15). - císlo(V) - ;
% císlo(V) :- cifra(X), cifra(Y), V is 10*X+Y, X > 0.
```

```
% n ciferné číslo s levou rekursí - císlo(15555). - císlo(V) - ;
císlo(V) :- císlo(X), cifra(Y), V is 10*X+Y, X > 0.
```

```
% n ciferné číslo s pravou rekursí - císlo(15555). - císlo(V) - ;
% císlo(V) :- cifra(Y), císlo(X), V is 10*X+Y, X > 0.
```

```
% násobení 2 čísel - ( krat(5, 5).
krat(X, Y) :- císlo(X), císlo(Y), V is X*Y, write('Výsledek: '), write(V).
```

```
% načte číslo z klávesnice -
zadej_císlo(X) :- write('Zadej číslo: '), read(X).
```

```
% zadáš číslo x, pak zadáš y a čísla se mezi sebou vynásobí - krat. 45. 78.
krat :- zadej_císlo(X), zadej_císlo(Y), krat(X,Y).
```

```
% nekonečný výpis čísel
% fail je neustále nastavený na false a já se snažím nabít hodnotu true -> což nenastane
% kontroluje zleva doprava
vypis_radu :- císlo(X), write(X), write(' '), fail.
```

```
hrana(1, 4).
hrana(1, 5).
hrana(2, 1).
hrana(2, 4).
hrana(3, 1).
hrana(3, 2).
hrana(3, 4).
hrana(4, 5).
```

```
% v komentářích za pomlčkou následuje co vepisovat do konzole - ukázky
% příkazy v konzoli nutno ukončovat tečkou! Jinak si prolog myslí, že číslo neskončilo
% chci příkazy v konzoli ukončit - quit. a program znovu přelož - Run as/Interpreted Single File
```

```
% nerekurzivní - cesta(2, 5) - cesta(2, X). ; ; ; ; no
cesta(X, Y) :- hrana(X, Y).
```

```
% rekurzivní
cesta(X, Y) :- hrana(X, Z), cesta(Z, Y).
```

```
% nalezne cestu z čísla do čísla a do X - cesta(2, 5, X) ; ; ;
cesta(X, Y, ([X, Y]) :- hrana(X, Y).
```

```
cesta(X, Y, [X | T]) :- hrana(X, Z), cesta(Z, Y, T).
```

```
% konečný cyklus, neboť graf je konečný - cestyl. no
cestyl :- cesta(2, 5, X), write(X), fail.
```

```
% tenhle zápis faktu, má vždy pravdivostní hodnotu true,
% proto místo no, vypíše yes (oproti předchozímu) - cestyl. yes
cestyl.
```

```
% findall: má 3 argumenty, hledá hodnoty X, jak? pomocí predikátu cesta(2, 5, X)
% a všechny různé hodnoty X bude ukládat do X, dostaneme tzv. seznam seznamů
```

```
% Cesty s velkým C = proměnná
% do konzole - cestyl. [[2, 1, 5], [2, 1, 4, 5], [2, 4, 5]] yes
cestyl2 :- findall(X, cesta(2, 5, X), Cesty), write(Cesty).
```

/*

Typicky druh ulohy pro PROLOG

Kacer, racek, kohout a jezek si jednou vysli do hospody. Druhý den si toho prilis nepamatovali. Vedeli pouze, ze:

- kazdý pil ten večer pouze jeden druh pití (pivo, zelenou, rum, víno)
 - jeden z nich spal na lavièce
 - ten, kdo pil rum, nespál doma ani na louce
 - kohout ani racek rum nikdy nepijou
 - rackovi bylo ráno velice spatně
 - kohout a jezek vedí, ze urcite nespali na louce
 - kaèer a jezek si pamatují, ze nespali na namestí
 - ten, kdo pil pivo, nespál doma ani na louce
 - kohout nikdy nepije pivo a kacer by se nikdy nenapil zelené
- Pomozte jim urcit, kdo co pil a kde spal, kdyz víte, ze kazdý spal jinde.

*/

zviratko(kacer).

zviratko(racek).

zviratko(kohout).

zviratko(jezek).

% X = zviratko

co_pil(X, pivo) :- zviratko(X), X \= kohout.

co_pil(X, zelenou) :- zviratko(X), X \= kacer.

co_pil(X, rum) :- zviratko(X), X \= kohout, X \= racek.

co_pil(X, vino) :- zviratko(X).

% X = zviratko, Y = piti

kde_spal(X, Y, lavicka) :- co_pil(X, Y).

kde_spal(X, Y, louka) :- co_pil(X, Y), Y \= rum, Y \= pivo, X \= kohout, X \= jezek.

kde_spal(X, Y, namesti) :- co_pil(X, Y), X \= kacer, X \= jezek.

kde_spal(X, Y, doma) :- co_pil(X, Y), Y \= rum, Y \= pivo.

ruzne(A1, A2, A3, A4) :- A1 \= A2, A1 \= A3, A1 \= A4, A2 \= A3, A2 \= A4, A3 \= A4.

kdo_kde(X1, Y1, X2, Y2, X3, Y3, X4, Y4):- kde_spal(X1, Y1, lavicka),

kde_spal(X2, Y2, louka),

kde_spal(X3, Y3, namesti),

kde_spal(X4, Y4, doma),

ruzne(X1, X2, X3, X4),

ruzne(Y1, Y2, Y3, Y4).

% spust - go.

go :- kdo_kde(Kdo1, Co1, Kdo2, Co2, Kdo3, Co3, Kdo4, Co4),

nl, write('VESELA ZVIRATKA:'), nl, write('-----'), nl,

write('Na lavicce spal '), write(Kdo1), write(', ktery pil '), write(Co1), write('.'),nl,

write('Na louce spal '), write(Kdo2), write(', ktery pil '), write(Co2), write('.'),nl,

write('Na namesti spal '), write(Kdo3), write(', ktery pil '), write(Co3), write('.'),nl,

write('Doma spal '), write(Kdo4), write(', ktery pil '), write(Co4), write('.').
