

Лекция

# Логические задачи

# Задача 1 (Зебра)

Пять человек различной национальности живут в пяти первых домах одной из улиц. Они являются представителями пяти различных профессий, и у каждого из них есть свои любимые напитки и животные, среди которых нет повторяющихся. Дома, в которых они живут, покрашены в различные цвета. Известно, что:

- Англичанин живет в красном доме.
- У испанца есть собака.
- Японец является художником.
- Итальянец пьет чай.
- Норвежец живет в первом доме слева.
- Владелец зеленого дома пьет кофе.
- Зеленый дом находится справа от белого.
- Скульптор разводит улиток.
- Дипломат живет в желтом доме.
- В доме посередине пьют молоко.
- Норвежец живет рядом с голубым домом.
- Скрипач пьет фруктовые соки.
- Лису держат в доме, соседнем с домом врача.
- Лошадь держат по соседству с домом дипломата.

Кто держит зебру, и кто пьет воду?

# Решение

Улица - список домов:

*[[нац, проф, цвет, животное, напиток],[...],[...],\_,\_]*

Предикат «справа»:

*next([X1,X2 | \_],X1,X2).*

*next([\_ | T],X1,X2):- next(T,X1,X2).*

Предикат «по соседству»:

*neighbour(Street,X1,X2):-next(Street,X1,X2).*

*neighbour(Street,X1,X2):-next(Street,X2,X1).*

# Предикат solve

`solve(Street,Z,W):-`

```
Street = [[norwegian,_,_,_],[_,_,blue,_,_],[_,_,_,_,milk],_,_],  
member([english,_,red,_,_],Street),  
member([spanish,_,_,dog,_,_],Street),  
member([japanese,painter,_,_,_],Street),  
member([italian,_,_,_,tea],Street),  
member([_,_,green,_,_,coffee],Street),  
next(Street,[_,_,white,_,_],[_,_,green,_,_]),  
member([_,sculptor,_,snails,_,_],Street),  
member([_,diplomat,yellow,_,_],Street),  
member([_,musician,_,_,juice],Street),  
neighbour(Street,[_,_,_,fox,_,_],[_,_,doctor,_,_,_]),  
neighbour(Street,[_,_,_,horse,_,_],[_,_,diplomat,_,_,_]),  
member([Z,_,_,zebra,_,_],Street),  
member([W,_,_,_,water],Street).
```

# Запросы

?- solve(X,Z,W).

X = [ [norwegian,diplomat,yellow,fox,water],  
[italian,doctor,blue,horse,tea],  
[english,sculptor,red,snails,milk],  
[spanish,musican,white,dog,juice],  
[japanese,painter,green,zebra,coffee] ]

Z = japanese

W = norwegian

## Пример 2 (Слесарь)

На одном заводе работали три друга: слесарь, токарь и сварщик. Их фамилии Борисов, Иванов и Семенов. У слесаря нет ни братьев, ни сестер. Он - самый младший из друзей. Семенов, женатый на сестре Борисова, старше токаря. Назовите фамилии слесаря, токаря и сварщика.

# Решение (1)

```
solve(B,I,S) :- permute([B,I,S],[slesar,tokar,svarshik]),not(contradiction([B,I,S])).
```

```
contradiction(V):- data(V,F,A,TF), logicalnot(TF,FT), data(V,F,A,FT).
```

```
logicalnot(true,false).
```

```
logicalnot(false,true).
```

```
data(_sister,[slesar],false).
```

```
data(_old,[tokar,slesar],true).
```

```
data(_old,[slesar,tokar],false).
```

```
data(_old,[svarshik,slesar],true).
```

```
data(_old,[slesar,svarshik],false).
```

```
data([B,_,_],sister,[B],true).
```

```
data([_,_S],old,[S,tokar],true).
```

```
data([_,_S],old,[tokar,S],false).
```

# Решение (2)

```
solve(B,I,S) :- permute([B,I,S],[slesar,tokar,svarshik]),not(contradiction([B,I,S])).
```

```
contradiction(V):- data(V,F,A,fact,TF), logicalnot(TF,FT), data(V,F,A,_,FT).
```

```
logicalnot(true,false).
```

```
logicalnot(false,true).
```

```
data(_,sister,[slesar],fact,false).
```

```
data(_,old,[tokar,slesar],fact,true).
```

```
data(_,old,[svarshik,slesar],fact,true).
```

```
data([B,_,_],sister,[B],fact,true).
```

```
data([_,_,S],old,[S,tokar],fact,true).
```

```
data(V,old,[X,Y],rule,TF):-
```

```
    data(V,old,[Y,X],fact,FT),
```

```
    logicalnot(FT,TF).
```



# Запросы

```
?-solve(B,I,S), write('Борисов- '), write(B), nl,  
    write('Иванов- '), write(I), nl,  
    write('Семенов- '), write(S).
```

Борисов- tokar

Иванов- slesar

Семенов- svarshik

### Пример 3 (Домино)

Алла, Галя, Лена и Марина играли в домино. Марина младше, чем Галя. Лена старше, чем любая из ее противниц. Марина старше, чем ее партнерша. Алле и Гале вдвоем больше лет, чем Лене и Марине вместе.

Кто с кем играл, как распределить девушек по возрасту?

# Решение

Пары: Лена – X  
Y – Z

Список возрастов (от старшего к младшему): Age = [X1,X2,X3,X4]

Предикат «старше»:

more(X,Y,[X|T]):-member(Y,T).

more(X,Y,[\_|T]):-member(Y,T),member(X,T),more(X,Y,T).

```
solve :- Age = [X1,X2,X3,X4], permute(Age,[alla,galia,lena,marina]),
remove(X,[alla,galia,marina],[Y,Z]),
more(galia,marina,Age),
more(lena,Y,Age), more(lena,Z,Age),
member(Pair,[[lena,X],[Y,Z]]), remove(marina,Pair,[MP]),
more(marina,MP,Age),
not(permute([alla,galia],[X3,X4])), not(permute([alla,galia],[X2,X4])),
write('lena-'),write(X),nl,
write(Y),write('-'),write(Z),nl,
write(Age).
```

# Запросы

?- solve.

lena-galia

alla-marina

[galia,lena,marina,alla]

## Лекция 8

# Логические задачи (часть 2)

# Простые числа

## Генерация простых чисел (способ 1)

```
prime(X):-not(not_prime(X)).  
not_prime(X):-X1 is X-1, for(Y,2,X1), 0 is X mod Y.  
  
primes(2,[2]).  
primes(N,[N|L]):- prime(N), !, N1 is N-1, primes(N1,L).  
primes(N,L):-N1 is N-1, primes(N1,L).
```

# Генерация простых чисел (способ 2)

## Решето Эратосфена



Решето Эратосфена — так называют следующий способ получения ряда простых чисел.

Из ряда чисел  
2, 3, 4, 5, 6, 7, 8, 9, 10, 11, ...  
вычеркивают кратные двум;  
4, 6, 8, 10, 12, ...  
— кратные трем:  
6, 9, 12, 15, ...  
— кратные пяти:  
10, 15, 20, 25, 30, ... и т.д.

Таким образом все составные числа будут просеяны, и останутся только простые числа 2, 3, 5, 7, 11, 13...

<del>1</del>	<u>2</u>	<u>3</u>	<del>4</del>	<u>5</u>	<del>6</del>	<u>7</u>	<del>8</del>	<del>9</del>	<u>10</u>
<u>11</u>	<del>12</del>	<u>13</u>	<del>14</del>	<del>15</del>	<del>16</del>	<u>17</u>	<del>18</del>	<u>19</u>	<del>20</del>
<del>21</del>	<del>22</del>	<u>23</u>	<del>24</del>	<del>25</del>	<del>26</del>	<del>27</del>	<del>28</del>	<u>29</u>	<del>30</del>
<u>31</u>	<del>32</del>	<del>33</del>	<del>34</del>	<del>35</del>	<del>36</del>	<u>37</u>	<del>38</del>	<del>39</del>	<del>40</del>
<u>41</u>	<del>42</del>	<u>43</u>	<del>44</del>	<del>45</del>	<del>46</del>	<u>47</u>	<del>48</del>	<del>49</del>	<del>50</del>

# Предикат *primes*

```
primes(N,L):-integers(2,N,IL), filter(IL,L).
```

```
% сгенерировать список
```

```
integers(Min,Max,[Min | L]):-Min=<Max, M1 is Min+1,  
    integers(M1,Max,L).
```

```
filter([],[]).
```

```
filter([I | L],[I | R]):-process(I,L,L1),filter(L1,R).
```

```
% вычеркивание числа
```

```
process(_,[],[]).
```

```
process(X,[A | L],[A | R]):-not(0 is A mod X), process(X,L,R).
```

```
process(X, [A | L],R):-0 is A mod X,process(X,L,R).
```



# Пример 4 (рыбы)

*Вернувшись домой с рыбной ловли, Боря, Толя и Володя так рассказывали дома о своих успехах:*

**Боря.-** Толя поймал только две рыбки.

Володя поймал на штуку больше, чем Толя.

Мы с Володией поймали на восемь рыбок больше, чем Толя.

Я наловил рыбы больше, чем Толя и Володя вместе.

**Толя.-** Володя не выудил ни одной рыбешки.

Боря говорит неправду, что я поймал только две штуки.

У меня и у Бори улов неодинаковый.

Володя и Боря вместе поймали тринадцать рыбок.

**Володя.-** Толя - самый удачливый из всех нас.

Я поймал на три рыбки больше, чем Боря.

Толя говорит неправду, что я ничего не поймал.

Боря и Толя наловили поровну.

*Тут уж ничего не поделаешь - настоящие рыбаки любят приврать.  
И каждый из наших рыболовов лишь два раза из четырех говорил правду. Сколько же рыбы наловил каждый из них?*

# Решение

```
solve(B,T,V) :-
```

```
    for(B,0,10),      for(T,0,10),      for(V,0,10),  
        check(b,[B,T,V]),      check(t,[B,T,V]),      check(v,[B,T,V]).
```

```
    for(A,A,_).
```

```
for(X,A,B) :- A < B, A1 is A+1, for(X,A1,B).
```

```
check(N,S):-
```

```
    remove(T1,[1,2,3,4],R), remove(T2,R,[L1,L2]),      say(N,T1,S),      say(N,T2,S),  
                                                                not(say(N,L1,S)),      not(say(N,L2,S)).
```

```
    say(b,1,[_,2,_]).
```

```
say(b,2,[_,T,V]):- V is T+1.
```

```
say(b,3,[B,T,V]):- B+V == T+8.
```

```
say(b,4,[B,T,V]):- B > T+V.
```

```
say(t,1,[_,_,0]).
```

```
say(t,2,[_,T,_]):- T \= 2.
```

```
say(t,3,[B,T,_]):- B \= T.
```

```
say(t,4,[B,_,V]):- B+V == 13.
```

```
say(v,1,[B,T,V]):- T > B, T > V.
```

```
say(v,2,[B,_,V]):- B == V-3.
```

```
say(v,3,[_,_,V]):- V > 0.
```

```
say(v,4,[BT,BT,_]).
```

# Запросы

?- solve(B,T,V).

B = 7

T = 9

V = 10

## Пример 5

По обвинению в ограблении перед судом предстали Иванов, Петров, Сидоров.

Следствием установлено следующее:

1) Если Иванов не виновен или Петров виновен, то Сидоров виновен.

2) Если Иванов не виновен, то Сидоров не виновен.

Виновен ли Иванов?

# Решение

```
assertion1(I,P,S):-I = nevin,!,S=vin.
```

```
assertion1(I,P,S):-P = vin,!,S=vin.
```

```
assertion2(I,P,S):-I = nevin,!,S=nevin.
```

```
assertion1(_,_,_).
```

```
assertion2(_,_,_).
```

```
solve(I):-
```

```
    L=[vin,nevin],
```

```
    member(I,L), member(P,L),    member(S,L),
```

```
    assertion1(I,P,S),
```

```
    assertion2(I,P,S).
```

# Запросы

?- solve(I).

I=vin.

## Пример 6

Три ученицы – Тополева, Берёзкина и Клёнова – посадили около школы три дерева: березку, тополь и клен. Причем не одна из них не посадила то дерево, от которого произошла ее фамилия. Узнайте, какое дерево посадила каждая из девочек, если известно, что Клёнова посадила не березку.

# Решение

```
klen_berez(bereza):-!,fail.  
klen_berez(_).
```

```
solve:-permute([T,B,K],[topol,bereza,klen]),  
not(T=topol),not(B=bereza),not(K=klen),  
klen_berez(K),write([T,B,K]).
```

?- solve.

[bereza,klen,topol].