

Решения на някои задачи от контролните.

1. Дефинирайте предикат, описващ връзката между собственик и име на котката, която притежава.
owns(ivan_petrov, pussy).
owns(maria_dimova, kitty).
 2. Дефинирайте предикат, описващ връзката между име на котка и нейните цвят и тегло
cat(pussy, black, 5.8).
cat(kitty, ginger, 4).
 3. Задайте цел, която намира белите котки.
?- cat(X, white, _).
 4. Задайте цел, която намира двама собственика на котки с еднакво име.
?- owns(X,Z), owns(Y,Z), X\=Y.
 5. Задайте цел, която намира котките на Мария Димова, които тежат повече от 10 кг.
?- owns(maria_dimova,X), cat(X,_,Y), Y>10.
 6. Задайте цел, която намира собствениците, които нямат нито една черна котка.
?- owns(X,_), \+ (owns(X,Y), cat(Y,black,_)).
-

- а) Дефинирайте с 2 факта предикат, който описва съответствието между продукт, в кое ястие се използва и в какво количество.
yastie('к. мляко', 'супа топчета', 20).
yastie('яйце', 'крем-карамель', 30).
 - б) Дефинирайте с 2 факта предикат, който описва съответствието между продукт колко калории съдържа (в 100 г) и какъв е неговият състав. Съставът да се представи чрез структура с аргументи количествата белтъчини, въглехидрати и мазнини (в 100 г).
prod('к. мляко', 62, s(3,1.2,3.6)).
prod('ориз', 312, s(18,26,5)).
 - в) Задайте цел, която извежда продуктите, които съдържат 15 г белтъчини.
?- prod(X,_,s(15,_,_)).
 - г) Задайте цел, която извежда състава (като цяла структура) на продуктите, които се използват за приготвяне на лимонов пай.
?- yastie(Z, 'лимонов пай', _), prod(Z,_,X).
 - д) Дефинирайте предикат, определящ кое ястие е калорично – което съдържа повече от 100 г. продукт с калоричност над 250 ккал.
cal_yastie(X):- yastie(Y,X,Z), prod(Y,C,_), C>250, Z>100.
 - е) Дефинирайте предикат, определящ продуктите с най-малко въглехидрати.
min(V):- prod(V,_,s(_,X,_)), \+ (prod(_,_,s(_,Y,_)), Y<X).
-

1. Дефинирайте предикат за пресмятане на стойността на функцията: $g(y) = \begin{cases} \frac{y-1}{y^2+1}, & y > 5 \\ 4, & y = 5 \\ 2|y| & y < 5 \end{cases}$

g(5,4):-!.
g(Y,R):- Y>5, !, R is (Y-1)/(Y*Y+1).
g(Y,R):- R is 2*abs(Y).

2. Дефинирайте предикат, който пресмята произведението $\prod_{i=1}^k \frac{2i+1}{(2i-3)^2}$ при дадено цяло $k > 1$.

$pr(1,3):- !$.

$pr(K,P):- K > 1, I \text{ is } K-1, pr(I,X), P \text{ is } X*(2*K+1)/((2*K-3)*(2*K-3)).$

1. Дефинирайте предикат, който вмъква символа s след всяка 0 в списък.

$ins([],[]).$

$ins([0|T],[0,s|T1]):- !, ins(T,T1).$

$ins([H|T],[H|T1]):- ins(T,T1).$

2. Дефинирайте предикат, който за даден числов списък $[c_1, c_2, c_3, c_4, \dots]$ пресмята сумата $3.c_2 + 3.c_4 + 3.c_6 + \dots$

$sum([],0).$

$sum([_],0).$

$sum([_,X|T],S):- sum(T,S1), S \text{ is } S1+3*X.$

Още няколко решени задачи.

1. Дефинирани са предикати $ul_grad(X,Y,Z)$, където X е име на улица, Y – име на град, а Z – дължина на улицата и $grad_str(X,Y)$, където X е име на град, а Y – името на страната, в която се намира градът.

Задайте цели, които извеждат:

а) улиците с дължина по-голяма от 5 км;

$?- ul_grad(X,_,Z), Z > 5.$

б) градовете и страните, в които има улица “Европа”;

$?- grad_str(X,Y), ul_grad('Европа',X,_.).$

в) два града, в които има улица с едно и също име;

$?- ul_grad(X,Y,_.), ul_grad(X,Z,_.), Y \neq Z$

г) страните, в които няма улица “Безименна”.

(Това означава да няма нито един град в страната, в който има улица “Безименна”).

$?- grad_str(.,X), \vee + (grad_str(Y,X), ul_grad('Безименна',Y)).$

2. Дефинирайте предикат, който намира i -ия член на рекурентната редица:

$$c_1 = 2, c_i = \frac{c_{i-1} + 3i - 8}{i - 1}, i > 1$$

$c(1,2):- !$.

$c(I,X):- I > 1, J \text{ is } I-1, c(J,Y), X \text{ is } (Y+3*I-8)/J.$

3. Дефинирайте предикат, който по даден списък от числа $[a_1, a_2, a_3, a_4, \dots]$, пресмята произведението: $(2.a_1 - a_2).(2.a_3 - a_4). \dots$

$pr([],1).$

$pr([X],P):- P \text{ is } 2*X. \text{ или } pr([_],1).$

$pr([X,Y|T],P):- pr(T,R), P \text{ is } R*(2*X-Y).$

4. Дефинирайте предикат, който за даден списък от числа $[a_1, a_2, a_3, a_4, \dots]$, пресмята сумата: $a_1^2.a_2+a_3^2.a_4+ \dots$

$sum([],0):- !$.

$sum([_],0):- !$.

$sum([X,Y|T],S):- sum(T,S1), S \text{ is } S1+X*X*Y.$

5. Дефинирайте предикат, който намира произведението на тези елементи от списък с цели числа, които събрани със следващия ги елемент се делят на 3.

```
prod([],1).  
prod([_],1).  
prod([X,Y|T],P):- (X + Y) mod 3 == 0, !, prod([Y|T],P1), P is X*P1.  
prod([_,Y|T],P):- prod([Y|T],P).
```

6. Дефинирайте предикат, който от даден списък $[a_1, a_2, \dots, a_n]$ съставя списък $[1, [a_1], 1, [a_2], \dots, 1, [a_n]]$

```
def([],[]).  
def([H|T],[1,[H]|L]):- def(T,L).
```

7. Дефинирайте предикат, който изтрива първото срещане на 4 в даден списък.

```
del4([],[]):- !..  
del4([4|T],T):- !.  
del4([X|T],[X|L]):- del4(T,L).
```

8. Дефинирайте предикат, който заменя с 0 всеки елемент на даден списък D, който не принадлежи на списък C (останалите елементи на D се запазват същите).

```
subst([],_,[]):- !..  
subst([H|T],C,[0|L]):- not member(H,C), !, subst(T,C,L).  
subst([H|T],C,[H|L]):- subst(T,C,L).
```

9. Дефинирайте предикат, който умножава по 3 всеки елемент на даден числов списък, пред който стои 1.

```
mult([],[]).  
mult([X|T],[X|L]):- mul3([X|T],L).  
  
mul3([X],[]):- !.  
mul3([1,X|T],[Y|L]):- !, Y is 3*X, mul3([X|T],L).  
mul3([_,X|T],[X|L]):- mul3([X|T],L).
```