

1 Колір графу (розфарбування):

```
color(red). color(green). color(blue).
```

```
coloring(A, B, C, D) :-  
    color(A), color(B), color(C), color(D),  
    A \= B, A \= C,  
    B \= C, B \= D,  
    C \= D.
```

```
% Приклад виклику:  
% ?- coloring(A, B, C, D).
```

2 Знайти квадрат числа:

```
square(X, Y) :- Y is X * X.
```

```
% Приклад виклику:  
% ?- square(4, Y). → Y = 16.
```

3 Перевірка, чи елемент є у списку:

```
member(X, [X|_]).  
member(X, [_|T]) :- member(X, T).
```

```
% Приклад виклику:  
% ?- member(3, [1,2,3,4]). → true.
```

4 Знаходження довжини списку:

```
list_length([], 0).  
list_length(_|T, N) :- list_length(T, N1), N is N1 + 1.
```

```
% Приклад виклику:  
% ?- list_length([a,b,c], N). → N = 3.
```

5 Фільтрація парних чисел зі списку (збираємо новий список):

```
even_number(X) :- 0 is X mod 2.
```

```
filter_even([], []).  
filter_even([H|T], [H|R]) :- even_number(H), filter_even(T, R).  
filter_even([H|T], R) :- \+ even_number(H), filter_even(T, R).
```

```
% Приклад виклику:  
% ?- filter_even([1,2,3,4,5], X). → X = [2,4].
```

6 Прості родинні зв'язки + пошук предка:

```
parent(john, mary).  
parent(mary, susan).  
parent(susan, kate).
```

```
ancestor(X, Y) :- parent(X, Y).  
ancestor(X, Y) :- parent(X, Z), ancestor(Z, Y).
```

```
% Приклад виклику:  
% ?- ancestor(john, kate). → true.
```

7 Сума елементів списку:

```
sum_list([], 0).  
sum_list([H|T], Sum) :- sum_list(T, Rest), Sum is H + Rest.
```

```
% Приклад виклику:  
% ?- sum_list([1,2,3], S). → S = 6.
```

8 Добуток елементів списку:

```
product_list([], 1).  
product_list([H|T], Product) :- product_list(T, Rest), Product is H * Rest.
```

```
% Приклад виклику:  
% ?- product_list([2,3,4], P). → P = 24.
```