?- livesWith(kim, x) ——— we want the X value

|| ①

livesWith(X, Y) :-
    livesAt(X, No, Addr),
    livesAt(Y, No, Addr).

kim = X' ✓
x = Y'

? livesAt(X', No, Addr), livesAt(Y', No, Addr).

⇒ ? livesAt(kim, No, Addr), livesAt(x, No, Addr)

|| ②

livesAt(kim, 37, 'Foostreet').

No = 37
Addr = 'Foo street'

③

livesAt(x, 37, 'Foo street')

|| ✓

livesAt(kim, 37, 'Foostreet')

X = kim ✓

<F8>
Backtrack to last match ('||')

④

livesAt(X, 37, 'Foo street')

||

livesAt(bill, 37, Foostreet)

X = bill ✓

?- append([1,2],[3,4],X) ← we want X

X || ✓

append(~~[]~~ , B , B)

try the rule

append([Head'|TailA'], B', [Head'|TailC']) :-
    append(TailA', B', TailC').

[1,2] = [Head'|TailA']

[3,4] = B'

X = [Head'|TailC']

so 1 = Head'   X = [1|TailC']

[2] = TailA'

?- append(TailA', B', TailC')   // the condition of the rule

⇒ ?- append([2], [3,4], TailC').

?- appd([2], [3,4], TailC').

× // ✓

appd(~~[]~~, B, B)

Try the rule appd([Head² / TailA²], B², [Head² / TailC²]) :-

appd(TailA², B², TailC²).

[2] = [Head² / TailA²]

[3,4] = B²

TailC' = [Head² / TailC²])

So

2 = Head²

[3 = TailA²

TailC' = [2 / TailC²]

?- appd(TailA², B², TailC²) // the condition of the rule

⇒ ?- appd([3, [3,4] ~~B²~~, TailC²).

?- append([], [3,4], Tail^2)

||

append([], B, B)

[3,4] = B

Tail^2 = B

X is a list;
the list starts with 1
the rest of the list is Tail'

Simplify the bindings:

$$\left.\begin{array}{l} X = [1 \mid Tail'] \\ Tail' = [2 \mid Tail^2] \\ Tail^2 = B \\ B = [3,4] \end{array}\right\} \Rightarrow \left.\begin{array}{l} X = [1 \mid Tail'] \\ Tail' = [2 \mid Tail^2] \\ Tail^2 = [3,4] \end{array}\right\} \Rightarrow$$

$$\Rightarrow \left.\begin{array}{l} X = [1 \mid Tail'] \\ Tail' = [2,3,4] \end{array}\right\} \Rightarrow X = [1,2,3,4]$$

the answer