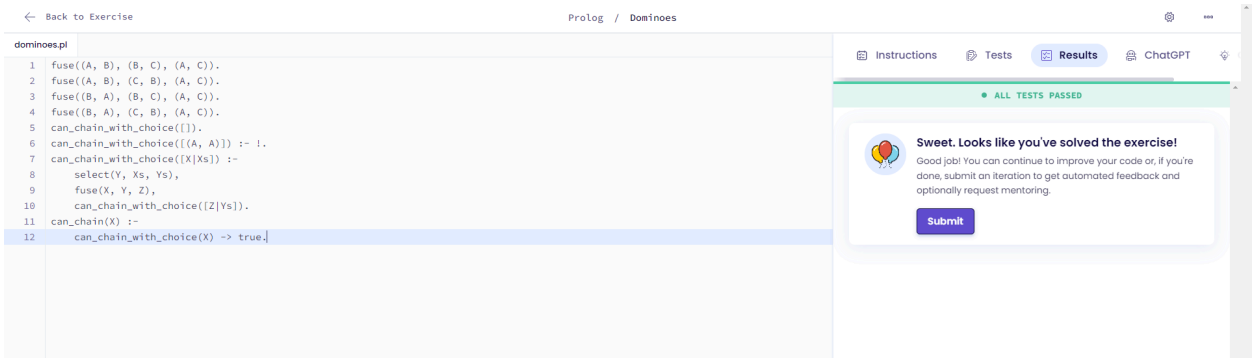
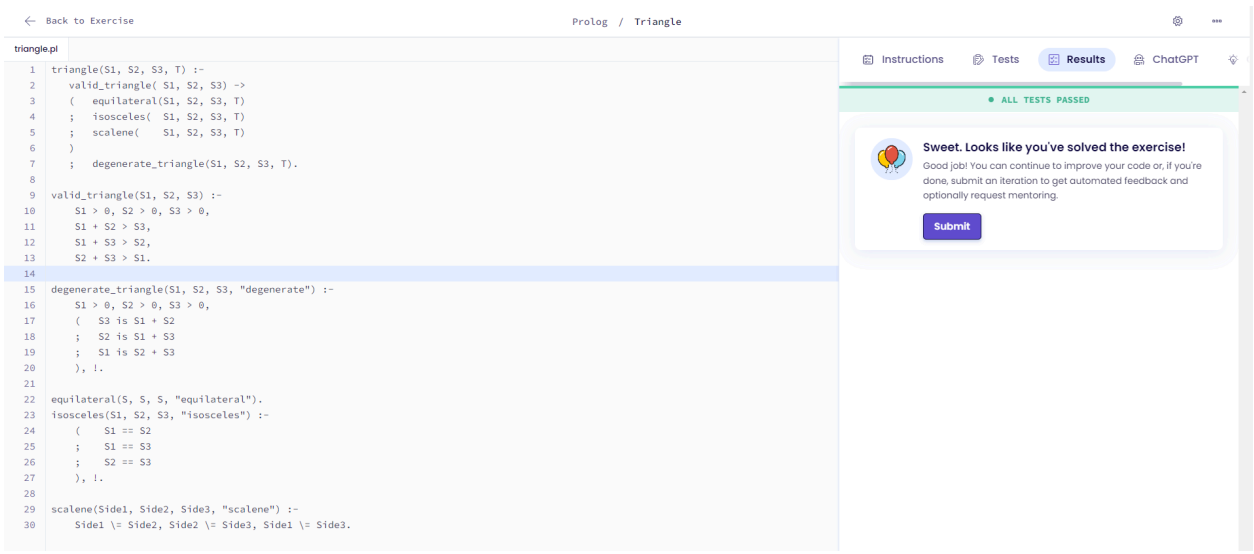
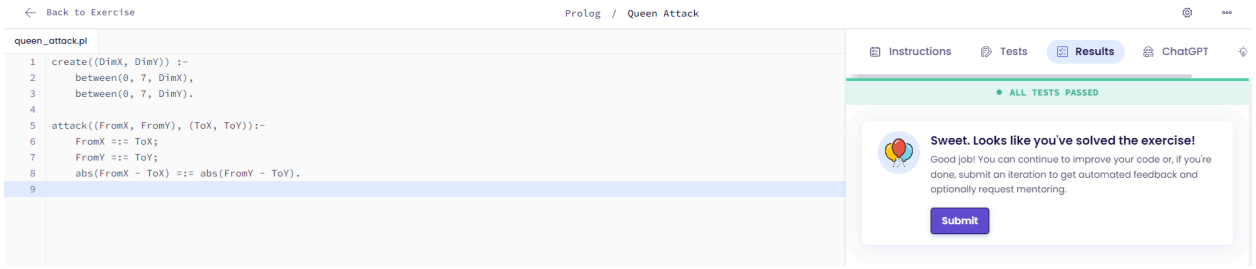
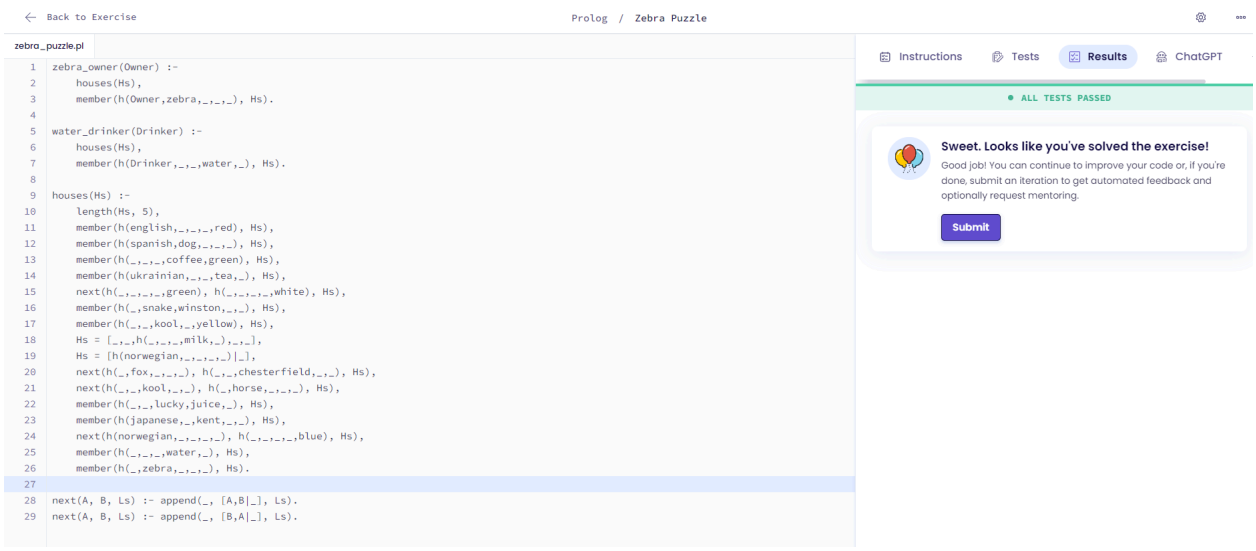

Back to Exercise
Prolog / Dominoes
dominoes.pl
fuse((A, B), (B, C), (A, C)).
fuse((A, B), (C, B), (A, C)).
fuse((B, A), (B, C), (A, C)).
fuse((B, A), (C, B), (A, C)).
can_chain_with_choice([]).
can_chain_with_choice([(A, A)]) :- !.
can_chain_with_choice([X|Xs]) :-
select(Y, Xs, Ys),
fuse(X, Y, Z),
can_chain_with_choice([Z|Ys]).
can_chain(X) :-
can_chain_with_choice(X) -> true.
Instructions
Tests
Results
ChatGPT
ALL TESTS PASSED
Sweet. Looks like you've solved the exercise!
Good job! You can continue to improve your code or, if you're done, submit an iteration to get automated feedback and optionally request mentoring.
Submit


Back to Exercise
Prolog / Triangle
triangle.pl
triangle(S1, S2, S3, T) :-
valid_triangle(S1, S2, S3) ->
(equilateral(S1, S2, S3, T)
; isosceles(S1, S2, S3, T)
; scalene(S1, S2, S3, T)
)
; degenerate_triangle(S1, S2, S3, T).
valid_triangle(S1, S2, S3) :-
S1 > 0, S2 > 0, S3 > 0,
S1 + S2 > S3,
S1 + S3 > S2,
S2 + S3 > S1.
degenerate_triangle(S1, S2, S3, "degenerate") :-
S1 > 0, S2 > 0, S3 > 0,
(S3 is S1 + S2
; S2 is S1 + S3
; S1 is S2 + S3
), !.
equilateral(S, S, S, "equilateral").
isosceles(S1, S2, S3, "isosceles") :-
(S1 == S2
; S1 == S3
; S2 == S3
), !.
scalene(Side1, Side2, Side3, "scalene") :-
Side1 \= Side2, Side2 \= Side3, Side1 \= Side3.
Instructions
Tests
Results
ChatGPT
ALL TESTS PASSED
Sweet. Looks like you've solved the exercise!
Good job! You can continue to improve your code or, if you're done, submit an iteration to get automated feedback and optionally request mentoring.
Submit


Back to Exercise
Prolog / Queen Attack
queen_attack.pl
create((DimX, DimY)) :-
between(0, 7, DimX),
between(0, 7, DimY).
attack((FromX, FromY), (ToX, ToY)):-
FromX =:= ToX;
FromY =:= ToY;
abs(FromX - ToX) =:= abs(FromY - ToY).
Instructions
Tests
Results
ChatGPT
ALL TESTS PASSED
Sweet. Looks like you've solved the exercise!
Good job! You can continue to improve your code or, if you're done, submit an iteration to get automated feedback and optionally request mentoring.
Submit


Back to Exercise
Prolog / Zebra Puzzle
zebra_puzzle.pl
zebra_owner(Owner) :-
houses(Hs),
member(h(Owner,zebra,_,_,_), Hs).
water_drinker(Drinker) :-
houses(Hs),
member(h(Drinker,_,_,water,_), Hs).
houses(Hs) :-
length(Hs, 5),
member(h(english,_,_,_,red), Hs),
member(h(spanish,dog,_,_,_), Hs),
member(h(_,_,_,coffee,green), Hs),
member(h(ukrainian,_,_,tea,_), Hs),
next(h(_,_,_,_,green), h(_,_,_,_,white), Hs),
member(h(_,snake,winston,_,_), Hs),
member(h(_,_,kool,_,yellow), Hs),
Hs = [_,_,h(_,_,_,milk,_),_,_],
Hs = [h(norwegian,_,_,_,_)|_],
next(h(_,fox,_,_,_), h(_,_,chesterfield,_,_), Hs),
next(h(_,_,kool,_,_), h(_,horse,_,_,_), Hs),
member(h(_,_,lucky,juice,_), Hs),
member(h(japanese,_,kent,_,_), Hs),
next(h(norwegian,_,_,_,_), h(_,_,_,_,blue), Hs),
member(h(_,_,_,water,_), Hs),
member(h(_,zebra,_,_,_), Hs).
next(A, B, Ls) :- append(_, [A,B|_], Ls).
next(A, B, Ls) :- append(_, [B,A|_], Ls).
Instructions
Tests
Results
ChatGPT
ALL TESTS PASSED
Sweet. Looks like you've solved the exercise!
Good job! You can continue to improve your code or, if you're done, submit an iteration to get automated feedback and optionally request mentoring.
Submit