# CST8503 Lab3 Prolog Programming

## Overview

We build on your experience from Labs 1 and 2 to write and trace more prolog programs.

When you have finished this lab exercise, you will have the ability to

* Write complex structures using Prolog syntax
* Predict how two different structures with variables will be matched
* Use a graphical tracing utility to watch your programs execute

## Complex Structures: points, line-segments, and polygons

Write the procedures described below, using the following functors:

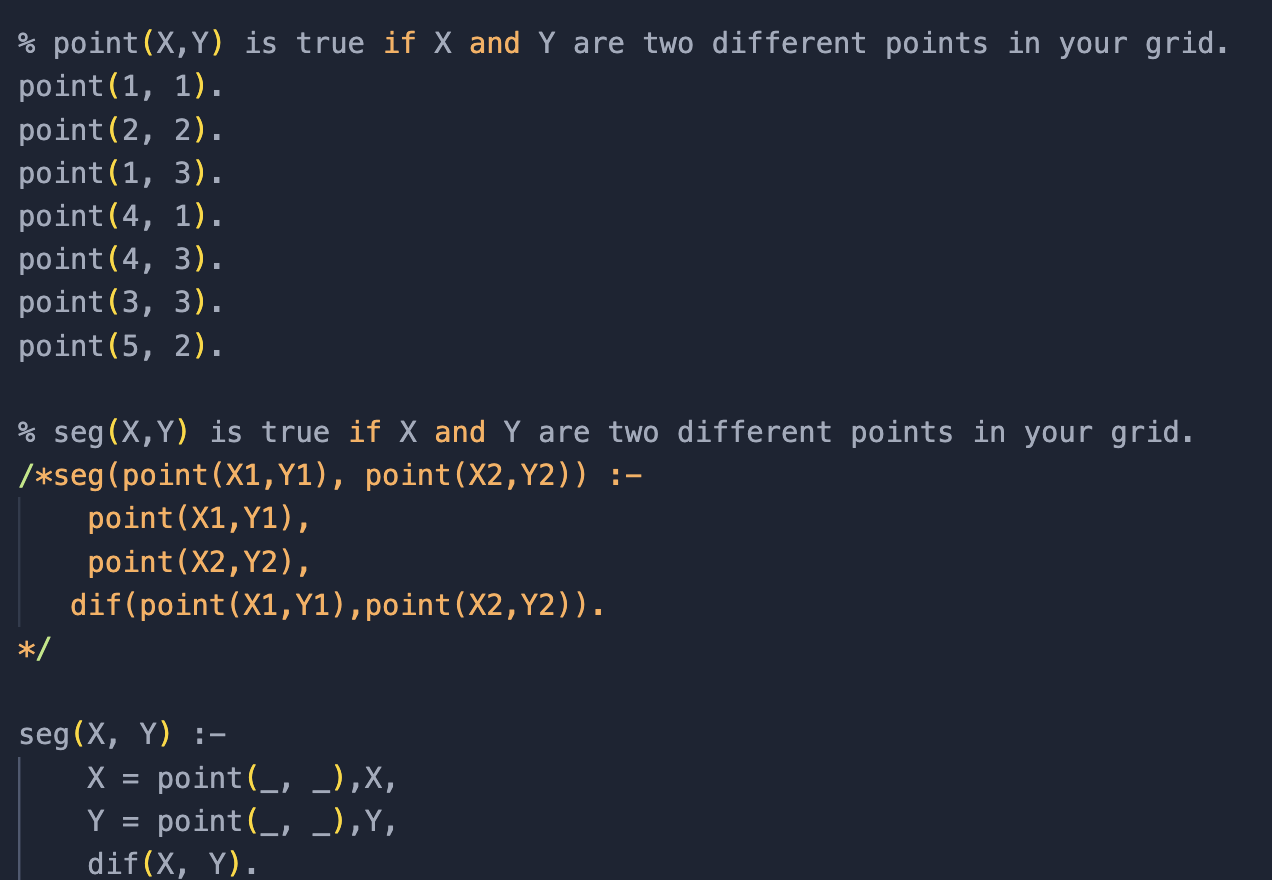
point/2

seg/2

triangle/3

rectangle/2

* Using the point/2 predicate, write down a representation five different points in an X-Y grid of integers, such that exactly two of the points have the same X-value, and none have the same Y-value.
* Write a seg(X,Y) procedure which is true if X and Y are two different points in your grid.



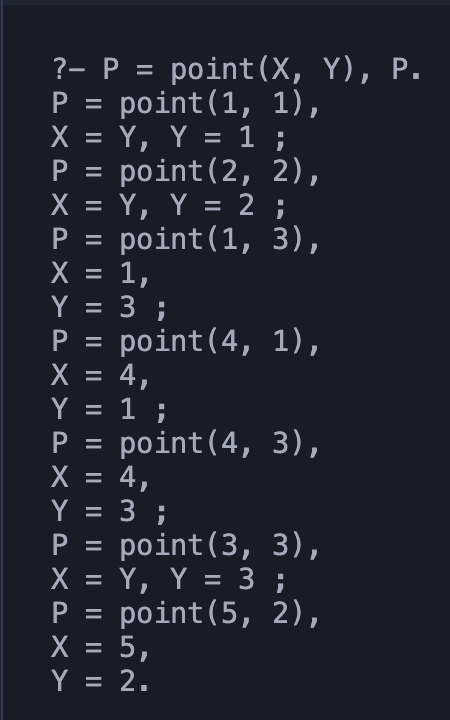
* + Test your procedure using pairs of specific points (you should execute at least one test that prints false, and one that prints true)

seg(point(1, 1), point(2, 2)).



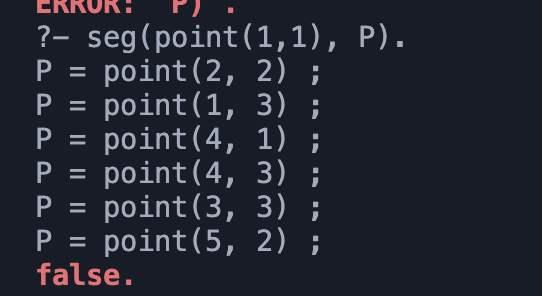
* + Try your procedure with points that are not explicitly defined by the point/2 predicate. Can you limit the procedure to work with only points that exist? (HINT: try the following query ?- P = point(X,Y), P. )

P = point(X,Y), P.

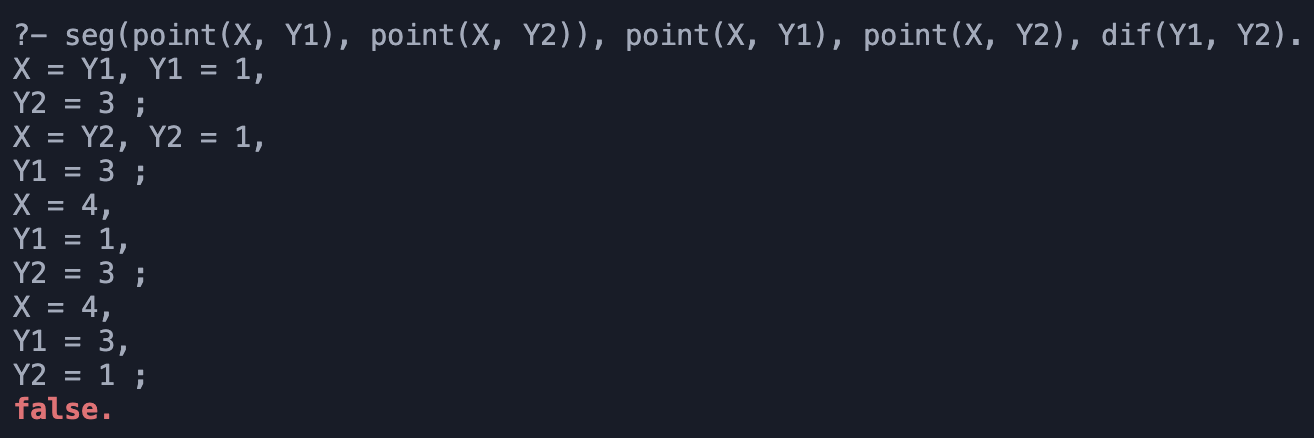


* + Use your procedure to find all line segments starting at a particular point (after each answer, type ";" to ask for more answers(use the previous hint)

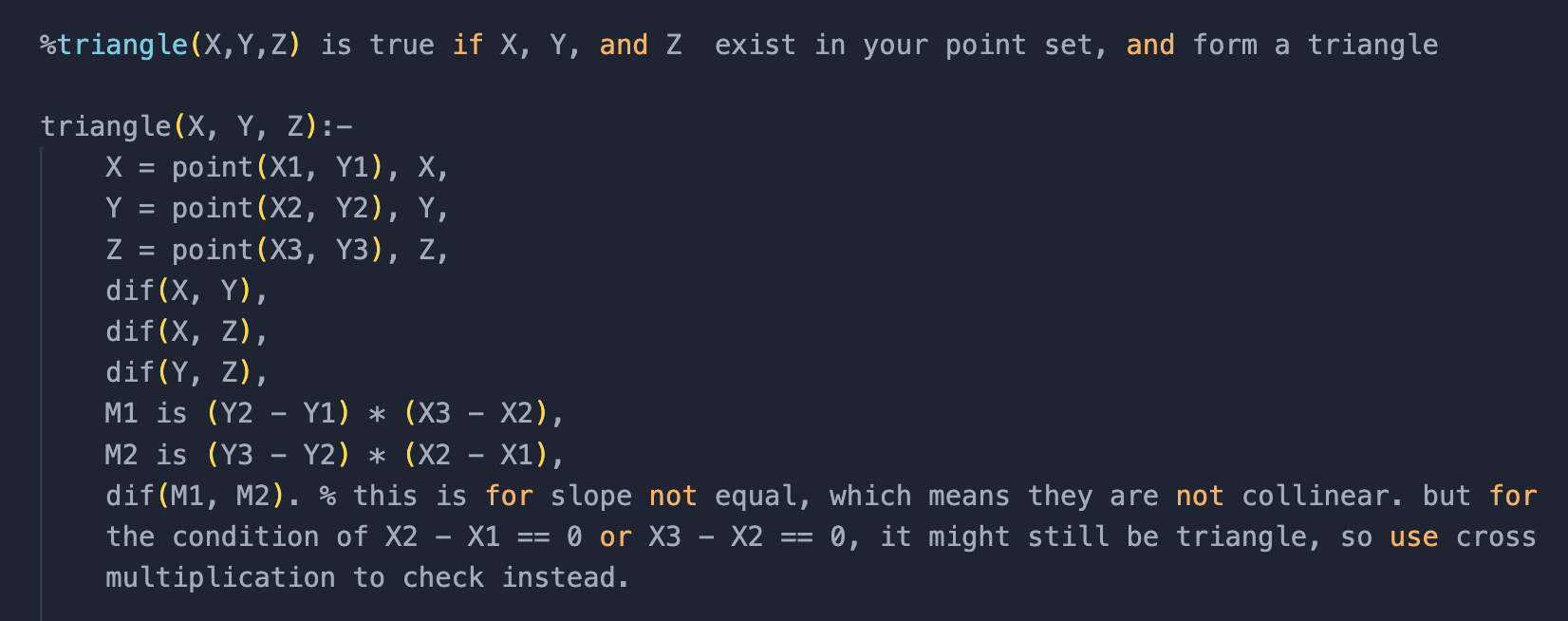
seg(point(1, 1), P).



* + Use your procedure to find all vertical line segments (note that two points with the same X value form a vertical line segment)



* Write a triangle(X,Y,Z) procedure which is true if X, Y, and Z exist in your point set, and form a triangle



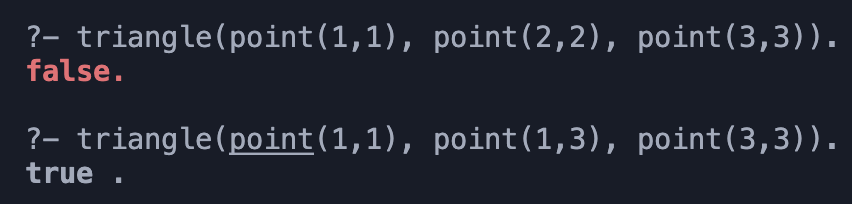
* + Test your procedure using triplets of specific points

?- :triangle(point(1,1), point(2,2), point(3,3)).

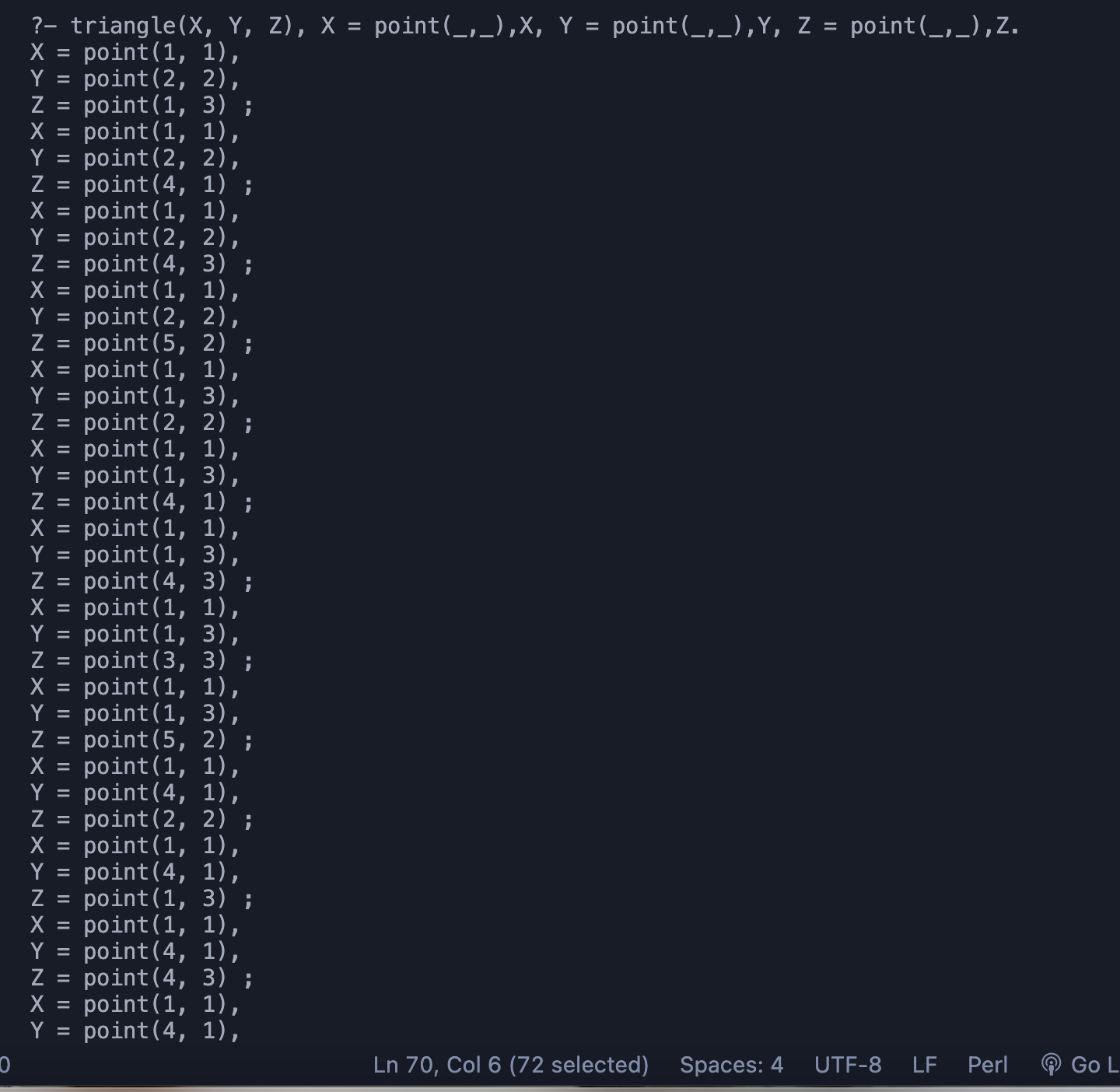
false.

?- triangle(point(1,1), point(1,3), point(3,3)).

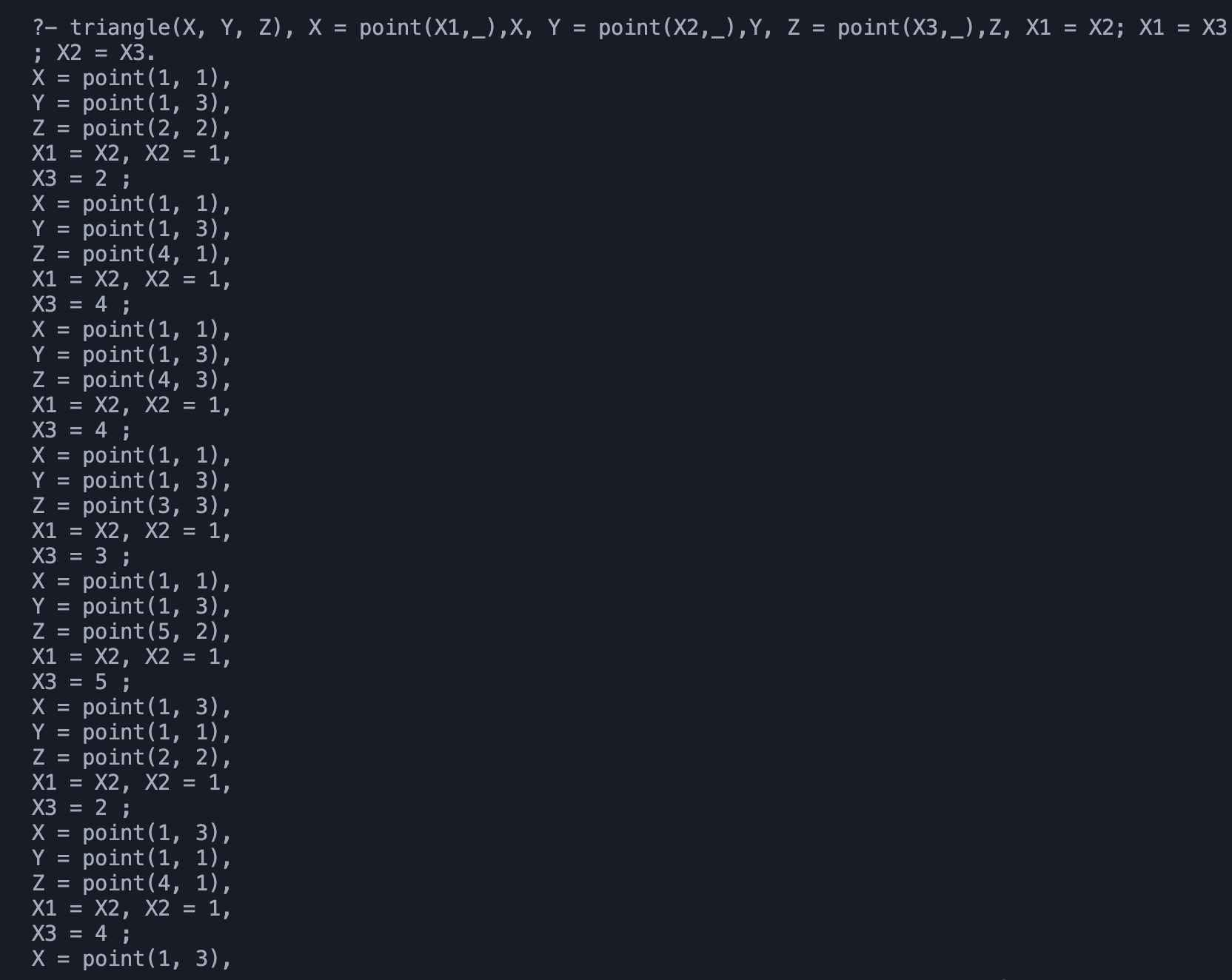
true.



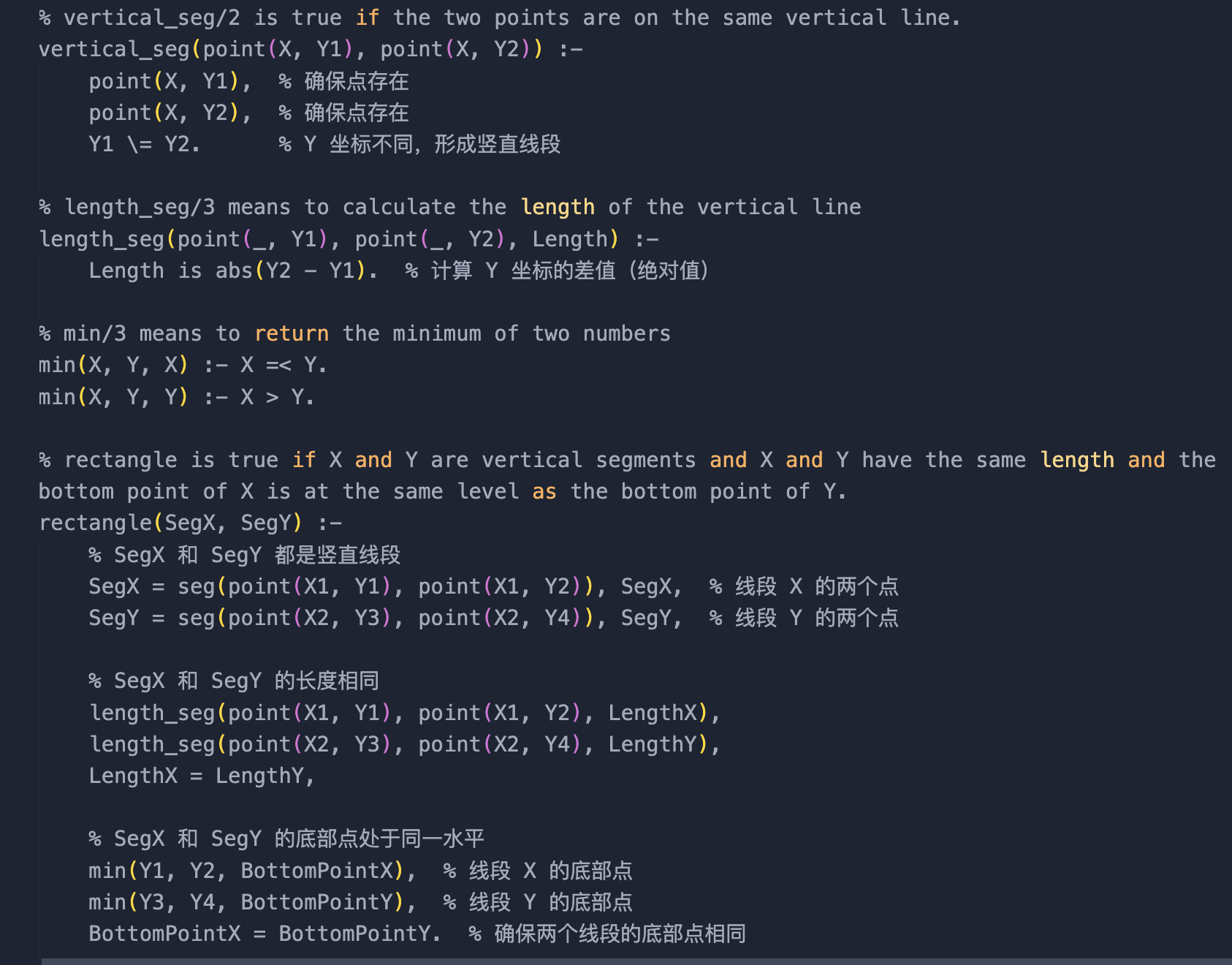
* + Use your procedure to find all triangles among your points



* + Use your procedure to find all triangles with a vertical side



* Write a rectangle(X,Y) procedure which is true if X and Y are different line segments such that
  + X and Y are both vertical
  + X and Y are the same length
  + The bottom point of X is at the same level as the bottom point of Y
  + Add points as necessary to test your procedure



% test cases

/\*

?- rectangle(seg(point(1, 1), point(1, 3)), seg(point(4, 2), point(4, 4))).

false.

?- rectangle(seg(point(1, 1), point(1, 3)), seg(point(4, 1), point(4, 3))).

true .

\*/

## Demonstration

* Show your program with complex structures
* Run your program with a query that succeeds
* Run your program with a query that fails
* Be prepared to explain to your instructor how two different structures with variables will be matched
* Show that you know how to use a graphical tracing utility to watch your program execute step by step