FP:  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
What is a predicate?

Any function that returns True or False (a boolean type).  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
Int се дели чрез div  
Double чрез /  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
Closure

When a function returns a lambda function,

we call that lambda function a closure.  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
a vector with two coordinates representing a point in 2D space:

type Point2D = (Int, Int) -- typedef std::pair<int, int> Point2D;

Now every function that accepts a Point2D will look like this:

helper :: Point2D -> Int -- instead of (Int, Int) -> Int  
  
helper (x, y) = x + y -- this function sums up the coordinates. Notice the pattern matching!

-- helper x = fst x + snd x -- only for 2D vectors  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_