```lua
-- Week 4, Activity 02
-- Match-time captures
--
-- Student: Abrantes Araújo Silva Filho


local lpeg = require "lpeg"
local pt = require "pt"


-- 1. Brief review of lpeg.P:
------------------------------

-- a) For match a specific pattern:
print(lpeg.P("hello"):match("hello 123"))                 --> 6

-- b) For match a specific number of characters:
print(lpeg.P(4):match("hello 123"))                       --> 5

-- c) For match a pattern with less than n characters:
--    lpeg.P(-n) = -lpeg.P(n)
print(lpeg.P(-9):match("hello 123"))                      --> nil
print(lpeg.P(-10):match("hello 123"))                     --> 1

-- d) To always match WITHOUT consuming anything:
print(lpeg.P(true):match("hello 123"))                    --> 1

-- e) To always fail WITHOUT consuming anything:
print(lpeg.P(false):match("hello 123"))                   --> nil

-- f) To create a grammar (we must use a table):
--     (not a complete example)
--    lpeg.P({...})


-- 2. MATCH-TIME CAPTURES:
--------------------------
-- lpeg.P could also receives a FUNCTION. This is called a
-- match-time capture, as explained below. See theese examples:

print(lpeg.P(print):match("hello 123"))                   --> hello 123     1
                                                          --> nil

print((3 * lpeg.P(print)):match("hello 123"))             --> hello 123     4
                                                          --> nil

-- The match-time capture always receives 2 arguments: the hole subject and the
-- current position. Because of this the "print" function is printing the hole
-- subject and the position. In the first example the match stays in the first
-- position, but in the second example the match consumed 3 characters and the
-- current position is 4.
--
-- Also note the "nil": this indicate that the matches FAIL. This is because the
-- match-time capture (the print function in those examples) has the power to
-- decide if the matches succeeds of fail, as the following:
--    a) IF the match-time function returns nil, false or no value, if FAILS;
--    b) IF the match-time function true, it SUCCEEDS, WITHOUT consuming;
--    c) IF the match-time function returns a number which is EQUAL or GREATER
--       than the current position, then it matches that number of characters.
```

```lua
print((3 * lpeg.P(function () return true end)):match("hello 123"))       --> 4

print((3 * lpeg.P(function (s, p) return p + 2 end)):match("hello 123"))  --> 6

-- Match-time functions are very powerfull because it receives the whole subject
-- and the position where it is the subject, and it can decide programmatically
-- whether if it's accept, if it's reject, and, if it's accept, how much of it
-- it accepts.
--
-- A special property that match-time function have is that LPeg ALWAYS CALL
-- the match-time function, whenever LPeg attemps to match the pattern. This is
-- completelly different from functions captures. See theese examples:

-- ((lpeg.P(1) / print)^0):match("123")                --> 1
                                                       --> 2
                                                       --> 3
                                                       --> 4

-- The 1, 2, 3 are the matches that have sent to print, and the 4 is the final
-- result of the whole match.

-- Now consider this:

-- ((lpeg.P(1) / print)^0 * "x"):match("123")          --> nil

-- In that case print is NEVER CALLED. It COULD BE CALLED because LPeg does not
-- specify whether it calls or doesn't call functions captures, except when it
-- needs the value of a function capture, but otherwise it's undefined whether
-- it calls it or not. But with a match-time capture (and because of this that
-- it is called a match-time) the function IS ALLWAYS CALLED regardless the
-- match succeeds or not. It is that way bacause the match-time function must
-- have the power to decide if it matches or not. This is seen here:

p = lpeg.P(function(s, p) print(p); return true end)
print(((1 * p)^0 * "x"):match("123"))                  --> 2
                                                       --> 3
                                                       --> 4
                                                       --> nil

-- The whole matches fail (nil), but the function was called every time.

-- Match-time captures (whith this property of being called every time that LPeg
-- tries to match a patter) is very important becaus it's used in several
-- situations, like error reporting and debugging.
```