Problem Set 6
COMP301 Fall 2021
Week 8: 22.11.2021 - 26.11.2021

Instructions:

- Submit your answers to the Blackboard PS6 assignment until November 27th Saturday, at 23.59.
- Please use the code boilerplate, which includes several tests for you to see if your code is correct.
- Submit your code and PDF file to BlackBoard as a single zip file *yourIDno_ username.zip*. (Example: *123456_ftokmak17.zip*)

Problem 1: Determine whether the PROC codes given below run without error or not. If they run, evaluate their results and if they not, state the reason of error.

a).
```
let x = 5
  in let y = proc (z) -(t, z)
     in let t = 5
        in (y 3)
```

b).
```
let x = 1
  in let y= proc (z) -(z, x)
     in let x = 7
        in let z = 5
           in (y x)
```

c).
```
let x = 9
  in let y = proc (z) (z x)
     in let x = 0
        in let t = proc (x) if zero?(x) then 5 else 3
           in (y t)
```

Problem 2. [1]: Extend the `PROC` to include procedures with multiple arguments and calls with multiple operands, as suggested by the grammar:

$$\begin{aligned} \mathit{Expression} &::= \texttt{proc}(\{\mathit{Identifier}\}^{*(,)})\ \mathit{Expression} \\ &::= (\mathit{Expression}\{\mathit{Expression}\}^{*}) \end{aligned}$$

Here is an example usage which evaluates to 1:

```
let f = proc(x, y) -(x,y)
in (f 5 4)
```

[1] EOPL p.80 Exercise 3.21

Problem 3: Now, extend the latest version of PROC language with a new feature named as *double*. Please follow the grammar below.

Expression ::= `double`(*Expression Expression*)
`[double-exp procedure val]`

Double gets two expressions as argument. If f is a procedure and x is a number, double(f x) should return same result with (f (f x)).

Assume that f function gets only one number as an argument and produces a num-val.

Example: `let f = proc (x) -(x,-1) in double(f 5)` -> 7

Note 1: Methods that need to be modified are highlighted inside the `PROC` language source code with some hints.

Note 2: You need to update the following files: `data-structures.rkt`, `interp.rkt` and `lang.rkt`.