Problem 1: Basics

Please circle True or False for the following statements:

Ruby uses a static type system

Procs and Codeblocks can be used interchangeably

nil is not an object in Ruby

In Ruby, types are associated with values

Ruby has built in support for Regular Expressions

Problem 2: Code Completion

(a) Higher Order Programming

Fill in the blanks so that 10 is printed. You cannot hard code blank 1 or blank 2. You must use x and the codeblock.

```ruby
1 def myfunc(x)
2    puts __BLANK_1__
3 end
4 myfunc(3){__BLANK_2__}

blank 1:

Blank 2:
```

(b) Creation

Fill in the blanks so that a is a Hash with a default value of an Array of size 3

```ruby
a = __blank_1__

Blank 1:
```
(c) Objects

Fill in the blank so that square has a class variable called length with the value of x

```ruby
class Square
  def initialize(x)
    __Blank_1__
  end
end
```

Blank 1:

(d) Regex

Fill in the blanks so that "Correct" is printed

```ruby
rxp = /__Blank_1__/  
line1 = "23 years of age"
line2 = "1 year of age"
if rxp =~ line1 && rxp =~ line2
  puts "Correct"
else
  puts "Failed"
```

Blank 1:

Problem 3: Coding

Write a method named procHash(hash). The argument hash is a hash from a numerical key to a Proc. For each key and proc pair, print out "RESULT is the result of Proc(KEY)" where KEY is the key and RESULT is the result of calling the proc associated with the key on said key.

```
Example:
procHash({1=>Proc.new{|x| x * 2}, 2=>Proc.new{|x| x + 3}}) prints out
2 is the result of Proc(1)
5 is the result of Proc(2)
```