1. (2 pts) Programming language (PL). For the following multiple choice questions, circle the letter(s) on the right corresponding to the best answer(s) to each question.

a. Which following term is not a PL programming paradigm?  
   A) imperative  B) functional  C) logical  D) hierarchical

b. Which following term is not a desirable PL attribute?  
   A) libraries  B) relational  C) verifiable  D) natural

2. (8 pts) Ruby. What is the output (if any) of the following Ruby programs? Write FAIL if code does not execute. Output “nil” for “puts x” when x is nil (as in Ruby 1.8.7), instead of outputting a blank line (as in Ruby 1.9.3).

   a.  
      ```ruby
      a = [ ]
      a["turtle"] = "terrapin"
      puts "turtle = #{a["turtle"]}"
      ```
      
      OUTPUT = FAIL

   b.  
      ```ruby
      a = { }
      a[1] = "tortuga"
      puts "turtle = #{a[1]}"
      puts "shell" if a[2]
      ```
      
      OUTPUT = turtle = tortuga

   c.  
      ```ruby
      if "tortoise" =~ /(tr|is)/
      puts "found #{1}"  
      else
      puts "missing #{1}"  
      end
      ```
      
      OUTPUT = found is

   d.  
      ```ruby
      a = { }
      a[1] = "tesudo"
      a["Chelonii"] = 2
      a.keys.each { |x| puts x }
      ```
      
      OUTPUT = 1 Chelonii
3. (5 pts) Write a Ruby method `find_course_num` that given a string `str`, uses regular expressions and back references to find the first course number in the string. A course number has 4 uppercase letters, followed by a 3 digit number. The method should return the course number as an integer value, or `nil` if no course number is found.

Examples:

```
find_course_num("CMSC131 CMSC330")    # returns 131
find_course_num("CMsC131 CMSC330")      # returns 330
find_course_num("CMsC131 CMS330")       # returns nil
```

```
def find_course_num(str)
    if str =~ /[A-Z]{4}(0-9)[3]/
        return $1.to_i
    else
        return nil
    end
end
```

Some helpful functions (not all need to be used)

```
<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>a.each { … }</td>
<td>apply code block to each element in array</td>
</tr>
<tr>
<td>puts b</td>
<td>print b followed by a newline</td>
</tr>
<tr>
<td>s.to_i</td>
<td>returns integer value for string s</td>
</tr>
<tr>
<td>n.to_s</td>
<td>returns string for integer n</td>
</tr>
</tbody>
</table>
```

4. (5 pts) Write a Ruby method `print_str_array` that given an array of strings, uses the Array.each method and a code block to print out each string on a separate line, prefixed by the index of the string in the array, separated by a single space.

Examples:

```
print_str_array ("["terrapin","tortoise","turtle"]")   # prints
0 terrapin
1 tortoise
2 turtle
```

```
def print_str_array(arr)
    idx = 0
    arr.each { |x| puts "#{idx} #{x}" ; idx += 1 }
end
```

Some helpful functions (not all need to be used)

```
<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>a.each { … }</td>
<td>apply code block to each element in array</td>
</tr>
<tr>
<td>puts b</td>
<td>print b followed by a newline</td>
</tr>
<tr>
<td>s.to_i</td>
<td>returns integer value for string s</td>
</tr>
<tr>
<td>n.to_s</td>
<td>returns string for integer n</td>
</tr>
</tbody>
</table>
```

```
def print_str_array(arr)
    idx = 0
    arr.each { |x| puts idx.to_s + " + x ; idx += 1 }
end
```

```
def print_str_array(arr) idx = (0..(arr.size-1)).each { |x| puts "#{x} #{arr[x]}" } end
def print_str_array(arr) arr.size.times { |x| puts "#{x} #{arr[x]}" } end
```