

CMSC 330 Summer 2016 Quiz #1

Name: _____

Instructions:

- Do not start this test until you are told to do so!
- You have 15 minutes for this quiz.
- This is a closed book exam. No notes or other aids are allowed
- For partial credit, show all of your work and clearly indicate your answers.
- Write neatly. Credit cannot be given for illegible answers.

1. (8 pts) What is the output (if any) of the following Ruby programs? Write FAIL if code does not execute.

- | | | |
|----|--|---|
| a. | <pre>a = Hash.new("avenue") a["turnpike"] = 48 puts a["avenue"] puts a["turnpike"]</pre> | Output:
avenue
48 |
| b. | <pre>arr = [3, 42, 16, 9] arr.sort arr.reverse! puts arr</pre> | Output:
9 16 42 3 |
| c. | <pre>if "grades.cs.umd.edu" =~ /^(.*)umd\.edu\$/ then puts "success" puts \$1 else puts "you tried" end</pre> | Output:
success
grades.cs. |
| d. | <pre>b = String.new("Equality") c = String.new("Equality") if c == b then puts "Structural #{b}" else puts "Physical #{c}" end</pre> | Output:
Structural Equality |

2. (6 pts) Create a method `sort_by_length` that takes in an array of strings as an argument and sorts them by their length in ascending order. The order between two or more strings of the same length does not matter. (Hint: If you're stuck, use a Hash) **Example**

Input: ["examples", "not", "I", "making", "do", "like"]

Example Output: ["I", "do", "not", "like", "making", "examples"]

String Sort Method:

```
def sort_by_length(lst)
  return lst.sort {|x,y| x.length <=> y.length}
end
```

Sorting with Hash:

```
def sort_by_length(lst)
  len_hash = Hash.new {|k,v| k[v] = []}
  lst.each {|word|
    len_hash[word.length] << word
  }

  res = []
  sorted_lens = len_hash.keys.sort
  sorted_lens.each {|len|
    res += len_hash[len]
  }
  return res
end
```

3. (6 pts) Come up with a function `process_courses` that takes in an array of UMD **undergraduate** course IDs as an argument and returns a hash that maps each department ID with an array associated course numbers. If you come across invalid strings, do not include them in your hash. Valid input consists of four uppercase letters followed by a three digit number where the first digit is less than or equal to 4. For partial credit you can just write a regex that accepts valid strings.

Example Input: ["CMSC330", "CMSC099", "ECON414", "engl101", "BMGT595", "WWE200"]

Example Output: {"CMSC" => ["330", "099"], "ECON" => ["414"]}

```
def process_courses(lst)
  res = Hash.new {|k,v| k[v] = []}
  lst.each {|course|
    if course =~ /^[A-Z]{4}([0-4]\d{2})$/ then
      res[$1] << $2
    end
  }
  return res
end
```