**Open Source Software — CSCI-4470 — Spring 2022**
**Test 1**
**March 5, 2021**

**SOLUTIONS**

**Name:**

image

**RIN#:**

**Instructions:**

- Clearly print your name, RCS ID (in all caps.) and your RIN at the top of your exam.
- This test is open book, open notes and open computer. You **m**ay not use the internet. Please turn off your wifi.
- There are **6 questions** on this test worth a total of **100 points**.

1. Give regex expressions for each of the following (15 pts)

   (a) `^[a-z]{3,6}[0-9]*@rpi\.edu`

   (b) `^(518|\(518\)\s*|518(\-| ))276(\-| )*[0-9]{4}$`

   (c) `110 Eighth St\.\N*Troy\N*NY\N*12180`

   (a) Given a list of strings, one per line, recognize an RPI email address. Note that an RPI email address has the form rcsid@rpi.edu, where `rcsid` is between 3 and 6 alphabetic characters followed by 0 or more digits. All other strings should not be recognized. (Note that `pattern{lower, upper}` recognizes between lower and upper repetitions of pattern. I.e. `a{2,3}` recognizes `aa` and `aaa`) . You can assume each email starts at the beginning of the line.

   Given:

   ```
   turnew2@rpi.edu
   re@rpi.edu
   turnerw2@rpi.edu
   turnew@rpi.edu
   ```

   The expression should recognize `turnew2@rpi.edu` and `turnew@rpi.edu`

   (b) RPI has a campus in Troy, NY with an area code 518 and a prefix 276. The last 4 digits are arbitrary. Write a regular expression to recognize any sequence of 10 digits that can be interpreted as a valid RPI phone number. It should handle 10 consecutive digits with the correct area code and prefix, it should handle the area code in parentheses, or it should handle area code, prefix and number separated by dashes, spaces or a combination of both, For example, the regex should recognize all of:

```
5182769999
(518) 276-9999
518-276-9999
518 276 9999
518276 9999
```

but not:

```
518(276)(9999)
518-276-09999
518)2769999
```

You can assume each phone number starts at the beginning of the line.

(c) Finally, recognize every line that has RPI's street address of `110 Eighth St., Troy, NY 12180` the strings "110 Eighth St.", " Troy", "NY" and "12180" must appear in order, but can be separated by an arbitrary number of other characters and whitespace so long as it is on the same line.

Your program should recognize:

```
        RPI 110 Eighth St. Troy, NY 12180
        RPI 110 Eighth St. Lally Building, Troy              NY    12180
```

But not:

```
    RPI 111 Eighth St. Troy, NY 12180
    RPI 110 Eighth Troy St. Lally Building, Troy              NY    12180
    RPI 12180 110 Eighth St. Lally Building, Troy              NY
```

2. For each of the following licenses, indicate whether the license is "Copyleft", "Weak Copyleft", "Permissive", or Not an Open Source/Free Software License. Circle the best answer. (14 pts.)

(a) **MPL**

A. Copyleft

B. **Weak Copyleft**

C. Permissive

D. Not an Open Source/Free Software License

(b) **QaPL**

A. Copyleft

B. Weak Copyleft

C. Permissive

D. **Not an Open Source/Free Software License**

(c) **BSD**

A. Copyleft

B. Weak Copyleft

C. **Permissive**

D. Not an Open Source/Free Software License

(d) **AGPL**

A. **Copyleft**

B. Weak Copyleft

C. Permissive

D. Not an Open Source/Free Software License

(e) **LGPL**

A. Copyleft

B. **Weak Copyleft**

C. Permissive

D. Not an Open Source/Free Software License

(f) **MIT**

A. Copyleft

B. Weak Copyleft

C. **Permissive**

D. Not an Open Source/Free Software License

(g) **EPL**

A. Copyleft

B. **Weak Copyleft**

C. Permissive

D. Not an Open Source/Free Software License

i. **MPL**

A. Copyleft

B. Weak Copyleft

C. Permissive

D. Not an Open Source/Free Software License

ii. **QaPL**

A. Copyleft

B. Weak Copyleft

C. Permissive

D. Not an Open Source/Free Software License

iii. **BSD**

A. Copyleft

B. Weak Copyleft

C. Permissive

D. Not an Open Source/Free Software License

iv. **AGPL**

A. Copyleft

B. Weak Copyleft

C. Permissive

D. Not an Open Source/Free Software License

v. **LGPL**

A. Copyleft

B. Weak Copyleft

C. Permissive

D. Not an Open Source/Free Software License

vi. **Apache**

A. Copyleft

B. Weak Copyleft

C. Permissive

D. Not an Open Source/Free Software License

vii. **LGPL**

A. Copyleft

B. Weak Copyleft

C. Permissive

D. Not an Open Source/Free Software License

3. Reconstruct the following page in MarkDown or Restructured Text . You may assume that the image is named `pexels-jennifer-murray-1067202.png` and is located in the same directory as this file. Write your answer on the next page. (18 pts):

   image

```
# About Butterflies

Picture
-------
![](pexels-jennifer-murray-1067202.png)

> Photo by Jennifer Murray from Pexels, this attribution is rendered as a block

Languages
---------
| Language | Word |
|:-----:|:-------:|
|English | butterfly|
| French| papillon|
|Spanish| mariposa|
|Turkish|kelebek|

Lifecycle
-----
* Egg
* Larva
* Pupa
* Adulthood
```

   or

```
=================
About Butterflies
=================

Picture
-------

.. image:: pexels-jennifer-murray-1067202.png

..

*Photo by Jennifer Murray from Pexels, this attribution is rendered as a blockq

..
Languages
```

```
---------
+----------+-----------+
| Language |    Word   |
+----------+-----------+
|  English | butterfly |
+----------+-----------+
|  French  |  papillon |
+----------+-----------+
|  Spanish |  mariposa |
+----------+-----------+
|  Turkish |  kelebek  |
+----------+-----------+

Lifecycle
-----
* Egg
* Larva
* Pupa
* Adulthood
```

(540,600)

4. Provide answers to the questions below. Your answers do not need to be long. A few sentences should be sufficient to capture the important points (15pts).

   (a) In *The Cathedral and the Bazaar*, what principle (number and quote) does Eric Raymond dub **Linus' Law**?

   **# 8 Given a large enough beta-tester and co-developer base, almost every problem will be characterized quickly and the fix obvious to someone.**

   (b) In *The Cathedral and the Bazaar*, what principle does Eric Raymond use to argue against writing code from scratch when existing software can be used or repurposed?

   **# 2 Good programmers know what to write. Great ones know what to rewrite (and reuse)**

   (c) Discuss the major differences between a **copyleft** and a **permissive** software license. At what point of the software development lifecycle does this apply?

   While copyleft licenses differ in what they consider **derivative works**, all **copyleft** licenses require some type of reciprocity in how derivative works are licensed. Typically, this requires that derivative works be licensed under the same or at least a compatible copyleft license. A **permissive** license generally gives broad latitude to how derivative works are licensed. These differences only come into effect when software is distributed.

5. Consider the following scenario. You have a fork of an open source library in github. You can assume that the original is at: `git@github.com:RCOS/project.git` and your fork is at `git@github.com:me/project.git`.

   Show the sequence of git commands to do the following:

   - Establish a local repository based on your fork
   - Establish remote links to the original github repository *(upstream)* and to your fork *(remote)*
   - Create a branch (*mybranch*) on your local repository

   Now assume you have a modified file *modified.txt* on your local repository complete the sequence by:

   - Get the changes to *modified.txt* added to *mybranch*
   - Apply the changes in *mybranch* and any changes made to the upstream master to your *main*
   - Make the main branch on your github fork consistent with your local main branch

   You must use the command line for all git operations. (18 pts)

   Write your git commands below

```
git clone git@github.com:me/project.git
git remote add origin git@github.com:me/project.git # Optional - clone should do it
git remote add upstream git@github.com:RCOS/project.git
git checkout -b mybranch
git add modified.txt
git commit -m "Changes" # message is arbitrary
git checkout main
git merge mybranch
git pull upstream main
git push remote main
```

   (540,460)

6. Consider build systems. Answer the following questions (23pts):

   (a) Write a `Makefile` that captures the relationship depicted in the depndency diagram below. (16 of 23 pts).

   image

   A few notes:

- Do not use shortcuts. Put in the commands to create each object file and each library, etc. explicitly in the file.
- The `Makefile` must be hand generated. Do not use a Makefile generator (such as cmake) to create the file.
- Make sure you have `clean` and `all` build targets.

**Write your Makefile on the next page.**

```
all: a d

clean:
    rm a d libe.a a.o b.o c.o d.o

a.o: a.c a.h c.h
    cc -c -o a.o a.c

b.o: b.c
    cc -c -o b.o b.c

c.o: c.c c.h
    cc -c -o c.o c.c

d.o: d.c
    cc -c -o d.o d.c

a: a.o libe.a
    cc a.o libe.a -o a

d: d.o
    cc d.o -o d

libe.a: b.o c.o
    ar qc libe.a b.o c.o
```

(540,650)

(b) Now consider the CMakeLists file below. Add commands to the file to generate 3 tests. (7 of 23 pts)

- Test 1 - Verify “main” runs
- Test 2 - Verify the usage information exists
- Test 3 - Verify that running main with 32 returns the value 18

Here is the output of running “main”:

```
>>> main
Usage: main number
```

```
>>> main 32
18
```

**CMakeLists.txt**

```
cmake_minimum_required(VERSION 3.0)
project(Main C)

add_library(shared SHARED shared/shared.c)

add_library(static STATIC static/static.c)

add_executable(main main.c helper.c)
target_link_libraries(main shared static)

install(TARGETS shared DESTINATION lib)
install(TARGETS main DESTINATION bin)
```

```
cmake_minimum_required(VERSION 3.0)
project(Main C)

add_library(shared SHARED shared/shared.c)

add_library(static STATIC static/static.c)

add_executable(main main.c helper.c)
target_link_libraries(main shared static)

install(TARGETS shared DESTINATION lib)
install(TARGETS main DESTINATION bin)

enable_testing()

# does the application run
add_test(NAME Runs COMMAND main)

# does the usage message work?
add_test(NAME Usage COMMAND main)
set_tests_properties(Usage
PROPERTIES PASS_REGULAR_EXPRESSION "Usage:.*number"
)

# does the program work correctly?
add_test(NAME Test1 COMMAND main 32)
set_tests_properties(Usage
PROPERTIES PASS_REGULAR_EXPRESSION "18"
)
```

Your additions. (Do not repeat the above lines.)

(540,370)