Gotchas: Files, Parsing XML, JSON, HTTP, Regular Expressions Nov 7.2015

Please pair with someone, collaborate and help each other. Each section lists points earned(100+ means superb). Note: For C++ users this will be significantly more challenging since most of the tasks lack native support expect for the file I/O task and regex tasks.

A. File Access and Count Letter Frequency – 30 points

Note: For JavaScript unless you're using Node.js just skip the problem because file-access is forbidden. But if you're using Node.js here's a link that might help - https://gist.github.com/Arahnoid/9925725

a. In your language create a program that opens a file in read-only mode. After reading the entire file, print a frequency table for each letter from A to Z. Ignore case, which means little 'a' considered the same as big 'A'. Ignore anything other than A to Z, i.e. periods, commas, etc. Assume the last line has a carriage return on it(for buffered line reading systems). Example screen output would be:

```
Letter, Frequency
A,432
B,255
C,148
D,201
E,978
...
Z,48
```

Helpful links:

Java/Android:
http://www.homeandlearn.co.uk/java/read_a_textfile_in_java.html
http://stackoverflow.com/questions/13151714/reading-text-file-into-a-char-array-in-java
C++: http://www.cplusplus.com/doc/tutorial/files/
http://stackoverflow.com/a/18398836
Swift: http://stackoverflow.com/a/24098149

Python: http://www.pythonforbeginners.com/files/reading-and-writing-files-in-python
http://stackoverflow.com/a/19909871

b. Next, change your program so that it opens a new file in write mode and outputs what it used to print onto the screen into the file.

B. Parse an XML File or String – 30 points

In your language find out if there's a special support parsing XML files. Here are some links that may help (some are untested, further research may be needed):

Java: http://stackoverflow.com/a/8408730
Android: http://www.sitepoint.com/learning-to-parse-xml-data-in-your-android-app/
C++: https://github.com/leethomason/tinyxml2
Swift: http://dubinski.org/wpis/easy-xml-parsing-in-swift/
Python: https://docs.python.org/2/library/xml.etree.elementtree.html
JavaScript: http://www.w3schools.com/xml/dom_intro.asp

a. Copy the following XML file into a file or string and include it in your project.

```
<message>HELLO!</message>
```

b. Note: Important! The debugger in IntelliJ for the DOMParser is broken. Debug manually by simply running and adding logging statements.

Can you parse the document and print out the word, HELLO!. If not, you may need to add a header(shown below) at the beginning of the string or file.

```
<?xml version="1.0"?>
```

c. Enter the XML below into a file... Traverse the tree and print all the

nodes whose value is divisible by 3. Try to use recursion to solve the problem.

```
<tree>50
      <tree>20
            <tree>10</tree>
      </tree>
      <tree>5
            <tree>30
                  <tree>3
                        <tree>21</tree>
                  </tree>
            </tree>
      </tree>
</tree>
```

C. Parse a JSON File or String – 30 points

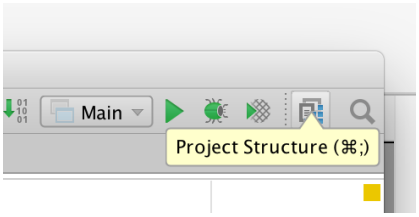
In your language find out if there's a special support for parsing JSON files. Here are some links that may help (some are untested, further research may be needed):

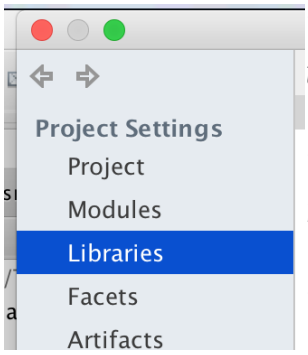
Java: http://crunchify.com/how-to-read-json-object-from-file-in-java/
Android: http://www.tutorialspoint.com/android/android_json_parser.htm
C++: https://github.com/open-source-parsers/jsoncpp
Swift: http://www.learnswift.io/blog/2014/7/30/parsing-json-in-swift
Python: http://docs.python-guide.org/en/latest/scenarios/json/
JavaScript: http://www.w3schools.com/json/json_eval.asp

Notes for Java users:

Java doesn't have built in support for JSON yet. It's supposed to be supported in Java 9, which hasn't come out yet. There's JSR 353(https://jsonp.java.net/), but I don't think it's part of the standard distribution yet. So, it's necessary to use 3rd party solutions.

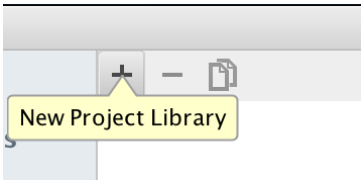
If you're using IntelliJ you can import the "com.googlecode.json-simple" dependency by clicking on the Project Structure icon on the upper-right corner of the IDE:



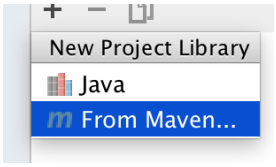


Then select the libraries option on the left panel in the dialog box:

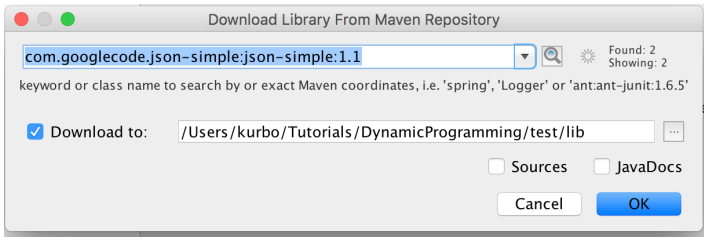
Then press on the "+" icon.



Then select "From Maven"



Type in "com.googlecode.json-simple" and press on the magnifying glass icon. Select the first open w/ 1.1 at the end. Check the download box. Press OK when ready.



Press OK on the next two dialog boxes. Then your project should work w/ the com.googlecode.json-simple external library.

a. Using the library for your language try and parse the following JSON string and print it out:

```
{ message:"Hello" }
```

b. Try a more complicated JSON string with an array inside and print everything out.

```
{ greetings:["Hello","Bonjour","Hola"] }
```

D. Fetch Data using HTTP GET – 30 points

In your language find out if there's a special support for issuing an HTTP GET and getting the response back. Here are some links that may help (some are untested, further research may be needed):

Java: http://crunchify.com/java-url-example-getting-text-from-url/
Android: http://www.javacodegeeks.com/2013/06/android-http-client-get-post-download-upload-multipart-request.html
C++: https://github.com/whoshuu/cpr
Swift: https://medium.com/swift-programming/learn-nsurlsession-using-swift-ebd80205f87c
Python: http://docs.python-requests.org/en/latest/user/quickstart/
JavaScript: http://stackoverflow.com/questions/247483/http-get-request-in-javascript

b. Now suse what you learned to write a program that execute an HTTP GET request to: "http://ip.jsontest.com/". It will return your ip address in JSON. Print out the string you got back.

c. Parse the JSON using the skills from section C, and print out only the ip address itself.

E. Practice using Regular Expressions – 30 points

Refer to the sheet and read over the basics for Regular Expressions. Figure out which string methods support regex in your language. Here are some links that may help (some are untested, further research may be needed):

Java/Android:
http://www.vogella.com/tutorials/JavaRegularExpressions/article.html
C++: http://www.informit.com/articles/article.aspx?p=2079020
http://www.cplusplus.com/reference/regex/
Swift: http://stackoverflow.com/a/26829118
https://forums.developer.apple.com/thread/19509
Python: https://docs.python.org/2/library/re.html

JavaScript: http://eloquentjavascript.net/09_regexp.html

a. Now using your knowledge of regular expressions. Open up a file using the knowledge gained in section A and filter out everything except for the letter a-zA-Z and spaces. Put the result in a string. Print the string to the screen.

b. Now using that string, only print out all the words that end in the letter 'e', by using regex to filter through all the words.