Part C

Write a program in a programming language of your choice, such as Python, Java, or C to do the following.

1. Request a URL from the user (or it could be entered on the command line when the program is invoked).

This URL could be just a domain name (www.uq.edu.au) or include the resource location (www.uq.edu.au/about).

You do NOT need to include http:// in this name, this can be assumed.

2. Request the designated web page, and analyse the response.

Print out a response to the screen, or to a text file, and ensure that you can capture the response so that it can be uploaded with your assignment source code.

The formatting of the response should follow the following template. The exact formatting is not so important but the information should be in the following Order.

HTTP Protocol Analyzer, Written by <Your name>, <your student-iD>

URL Requested: xxxxxxxxxxxxxxxxxx

IP Address, # Port of the Server: xxx.xxx.xxx.xxx , yyyy

IP Address # Port of this Client: xxx.xxx.xxx.xxx , yyyy

Reply Code: 304

Reply Code Meaning: Not Modified.

Date: Fri, 02 Mar 2018 12:01:02 AEST (please convert times to AEST if they are in GMT, otherwise leave them as they are)

Last-Modified: similar format (if appropriate to the response and supplied, otherwise just leave out, or write "not specified" or similar)

Content-Encoding: (if appropriate to the response, you should advertise at least compress, deflate and gzip accepted, if no content-encoding response, leave out or write "none")

Moved to: (if appropriate to the response)

Optional Extension

For cases where the file has moved permanently (301) or temporarily (302), and the next destination is also a http:// URL, then look up the new location and repeat the above information. Keep doing this for multiple redirections. If the next destination is https:// then you won't be able to open it directly with a normal socket.

Unmarked Challenge.

If you find this assignment easy and wish to extend yourself, you can try doing a similar program for https:// URLs.. No extra marks, and you'll need to explore yourself how to use the secure socket libraries.

LIBRARY RESTRICTIONS

You should use socket libraries directly to open the client TCP connection, and communicate between client and server.

You should NOT use higher level libraries or packages or programs which retrieve data from HTTP servers, such as the "Requests" library or the "urllib" library, or "http" package or similar.

You CAN use text-parsing functions which are part of "urllib", i.e. the "urllib.parse" module.

HAND IN:

1. Source Code for your program.

2. Your output for the following URLs: (note some are abc.net…, some abc.com…)

    www.csiro.au/awap/

    abc.net.au

    www.abc.com.au/news/sport

    www.abc.net.au/missing

    www.netbank.com

Marking Scheme: (Total Marks /100)

30 Marks: Code appears to run producing output that shows program can connect to a webserver with response 200. (for these marks, it is sufficient to simply echo the response from the server)

30 Marks: Code correctly shows responses in the correct order for responses 200, 301, 302, 304, 404 (you probably won't see any 304 codes unless you explicitly try to generate these, which is not necessary)

40 Marks: Extension Successfully Implemented

NOTE: The assignments are NOT auto-marked, so you don't need to be character perfect in the responses, but you should have the indicated information in the same order.