Part I: Python

A. Explain how the Python program extracts the web links from the HTML code of the "Current Estimates," found in web links section.

Python utilizes the BeautifulSoup library to parse the html code for the "Current Estimates" webpage. The html tag <a> defines a hyperlink, which is used to link from one page to another. By creating a BeautifulSoup loop for all <a> tags, Python can extract all web links from the "Current Estimates" webpage.

B. Explain the criteria you used to determine if a link is a locator to another HTML page. Identify the code segment that executes this action as part of your explanation.

The purpose of the <a href> attribute is to specify the URL of an external webpage. Therefore, I used this to my advantage to locate links that directed to external HTML pages [1,2].

```
#Collect all web links that direct to an html page
tags = soup.find_all(lambda tag: tag.name == 'a' and tag.get('href') and tag.text)
```

C. Explain how the program ensures that relative links are saved as absolute URLs in the output file. Identify the code segment that executes this action as part of your explanation.

Utilizing an IF/ELSE statement, I created a new list consisting of absolute URLs by iterating through the links set () and appending the links that started with a '/' with the absolute URL prefix [6].

```
#relative links are converted to absolute URLs
url_list = links
url_update =[]
for link in url_list:
    if link.startswith('/'):
        url_update.append('https://www.census.gov' + link)
    else:
        url_update.append(link)
```

D. Explain how the program ensures that there are no duplicated links in the output file. Identify the code that executes this action as part of your explanation.

To ensure there were no duplicate links in the final list, I removed all internal links and links that ended with an "/". I then used the set() operator to only returns unique elements, thus ensuring no duplicate links are in the output file [3].

```
#find all web links & add to set to eliminate duplicate records
links= set()
for item in soup.find_all(lambda tag: tag.name == 'a'and tag.get('href') and tag.text):
    links.add(item.get('href'))
```

```
#remove all internal links from dataset
for item in set(links):
    if item.startswith('#'):
        links.remove(item)

#remove the "/" at the end of links to prevent duplication
for item in set(links):
    if item.endswith('/'):
        links.remove(item)
```

E. Provide the Python code you wrote to extract all the unique web links from the HTML code of the "Current Estimates" (in the web links section), that point out to other HTML pages.

```
#find all web links & add to set to eliminate duplicate records
links= set()
for item in soup.find_all(lambda tag: tag.name == 'a'and tag.get('href') and tag.text):
    links.add(item.get('href'))

#remove all internal links from dataset
for item in set(links):
    if item.startswith('#'):
        links.remove(item)

#remove the "/" at the end of links to prevent duplication
for item in set(links):
    if item.endswith('/'):
        links.remove(item)

#relative links are converted to absolute URLs
url_list = links
url_update =[]
for link in url_list:
    if link.startswith('/'):
        url_update.append('https://www.census.gov' + link)
    else:
        url_update.append(link)
```

F. Provide the HTML code of the "Current Estimates" web page scrapped at the time when the scraper was run and the CSV file was generated.

File Name: census_html.txt

G. Provide the CSV file that your script created.

File Name: census_url.csv

H. Test your script and provide a screenshot of the successfully executed results.

```
'https://www.usa.gov',
'https://www.census.gov/topics/income-poverty/saipe.html',
'https://www.census.gov/about/contact-us/social_media.html',
'https://www.census.gov/topics/public-sector/redistricting-data.html',
'https://www.census.gov/data/tables/2017/demo/popest/total-housing-units.html',
'https://www.census.gov/blogs',
'https://www.census.gov/topics/public-sector/stories.html',
'https://www.census.gov/geography/education.html',
'https://www.census.gov/library/publications/2010/demo/p25-1138.html',
'https://www.census.gov/topics/economy/economic-census.html',
'https://www.census.gov/topics/business/services.html',
'https://www.census.gov/topics/employment/work-from-home.html',
'https://www.census.gov/topics/income-poverty/news-updates.html',
'https://www.census.gov/topics/health/working-papers.html',
'https://www.census.gov/topics/population/veterans.html',
'https://www.census.gov/topics/health/publications.html',
'https://www.census.gov/data/data-tools/quickfacts.html',
'https://www.census.gov/topics/health/disability.html',
'https://www.census.gov/topics/income-poverty/well-being.html',
'https://www.census.gov/programs-surveys/popest/about/special-census.html',
'https://www.census.gov/topics/health/news.html',
'https://www.census.gov/topics/income-poverty/data.html',
'https://www.census.gov/topics/international-trade/schedule-b.html',
'https://www.census.gov/newsroom/stories.html',
'https://www.census.gov/topics/families/stories.html',
'https://www.census.gov/programs-surveys/popest/about/fscpe.html',
'https://www.census.gov/topics/international-trade/news.html',
'https://www.census.gov/newsroom/facts-for-features.html',
'https://www.census.gov/topics/public-sector/congressional-apportionment.html',
'https://www.census.gov/programs-surveys/popest/guidance-geographies.html',
'https://www.census.gov/topics/education/data.html',
'https://www.census.gov/topics/housing/stories.html',
'https://www.census.gov/topics/families/children.html',
'https://www.census.gov/topics/employment/news.html',
```

Reference:

1. Extract links from webpage (BeautifulSoup). (n.d.). Retrieved from https://pythonspot.com/extract-links-from-webpage-beautifulsoup/

2. Extract links from webpage (w3schools). (n.d.). Retrieved from https://www.w3schools.com/tags/att_a_href.asp

3. Extract links from webpage (docs.python.org). (n.d.). Retrieved from https://docs.python.org/2/library/sets.html

4. Extract links from webpage (github.com). (n.d.). Retrieved from https://github.com/lorien/awesome-web-scraping/blob/master/python.md#url-and-network-address-manipulation

5. Extract links from webpage (reddit.com). (n.d.). Retrieved from https://www.reddit.com/r/learnpython/comments/2mmphx/saving_beautifulsoup_output_to_txt_file/

6. Consultation with Course Instructor, Brandon Vaughn [Telephone interview]. (2018, August 29).