Name: Sung Hwan Justin Yoon
Course: Programming in Python- C996

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

The python program extracts web links by requesting to open an url using the urllib library. The html code from the request is parsed using BeautifulSoup. This library parses through the code and finds the reference links that are found in the websites.

In [1]:

```
#import necessary libraries
from bs4 import BeautifulSoup
import urllib
import pandas as pd
```

In [2]:

```
#open the url and find all links
response = urllib.request.urlopen("https://www.census.gov/programs-surveys/popest.html")
soup = BeautifulSoup(response,from_encoding=response.info().get_param('charset'))
```

In [3]:

```
#save the html_code to html_code.txt
with open('html_code.txt','w',encoding='utf-8') as html_code:
    html_code.write(str(soup))
```

In [4]:

```
#view the first 500 characters of the html code
print(soup.prettify()[:500])
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-s
trict.dtd">
<html lang="en" xml:lang="en" xmlns="http://www.w3.org/1999/xhtml">
 <head>
  <!--[if lt IE 9]><meta http-equiv="X-UA-Compatible" content="IE=EmulateIE8"  /><![endif]-->
  <!--[if gte IE 9]><meta http-equiv="X-UA-Compatible" content="IE=edge"> <![endif]-->
  <meta content="text/html; charset=utf-8" http-equiv="Content-Type"/>
  <link href="/etc/designs/census/bootstrap.css" rel="styleshe
```

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.

To determine if the link was a locator to another html page, BeautifulSoup was used to find lines with the 'a' tag (which stands for anchor/hyperlinks) and those that were "href" (which stands for hypertext references).

In [5]:

```
#create a list called web_list and append all links with <a> that are "href"
web_list = []
for link in soup.find_all('a', href=True):
    web_list.append(link.get('href'))
```

In [6]:

```
#create a series with the webslist
web_series = pd.Series(data = web_list)
```

In [7]:

```
#15 weblinks found in the the scraped html
web_series.sample(n= 15)
```

Out[7]:

```
15         https://www.census.gov/topics/population.html
51                 https://www.census.gov/EconomicCensus
129     https://www.census.gov/programs-surveys/cps.html
178                   /programs-surveys/popest/news.html
59     https://www.census.gov/programs-surveys/survey...
133    https://www.census.gov/programs-surveys/poppro...
122    https://www.census.gov/programs-surveys/decenn...
171                               /programs-surveys.html
215                       https://www.census.gov/privacy
113            https://www.census.gov/library/audio.html
162                                             #content
148               https://www.census.gov/about/what.html
164                   /programs-surveys/popest/data.html
90     https://www.census.gov/topics/population/hispa...
106       https://www.census.gov/data/related-sites.html
dtype: object
```

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

The program ensured that relative links were saved as absolute URL's by using regular expressions. The regular expression that was used is shown in the code below. The string that was extracted from the array of links was made to only follow a pattern of "https://----.----.-----" (---- symbolizing any word/charcs). This meant that anything after ".com" or ".gov" or any other domain name was not extracted.

In [8]:

```
#extract using the websites that follow the regex pattern, and drop those that do not.
unique_website = web_series.str.extract('(https://[\w]+.[\w]+.[\w]+)').dropna()
```

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 that there were duplicates in the output file, the pandas method unique() was used. This method only gives the unique values of a pandas array. The code is shown below.

In [9]:

```
#get unique websites only.
unique_website = unique_website[0].unique()
```

In [10]:

```
#show the unique website in an array
#Notice how index 2 is not formatted the same as the others.
unique_website
```

Out[10]:

```
array(['https://www.census.gov', 'https://www.facebook.com',
       'https://twitter.com/uscensusbureau', 'https://www.linkedin.com',
       'https://www.youtube.com', 'https://www.instagram.com',
       'https://www.commerce.gov', 'https://www.usa.gov'], dtype=object)
```

In [11]:

```
#change index 2 (https://twitter.com/uscensusbureau to https://twitter.com)
unique_website[2] = 'https://twitter.com'
```

In [12]:

```
#write to a text file the unique websites
with open('unique_website.txt', 'w') as file:
    for website in unique_website:
        file.write("{}\n".format(website))
```

In [13]:

```
#print working directory to find the saved files
%pwd
```

Out[13]:

```
'C:\\Users\\jshyo\\Desktop\\WGU\\practice'
```