Homework #4

Due: November 27, 2023 **Points:** 100

This exercise has you query the PubMed database for a list of publications related to a keyword. Although we won't do it here, the list of publication numbers you get back can then be turned into a list of papers with a second query to the PubMed database.

To access the PubMed database, go to the URL below, replacing *keyword* with the keyword you want to search for, and *num* the number of publications you would like returned:

```
https://eutils.ncbi.nlm.nih.gov/entrez/eutils/esearch.fcgi?
     db=pubmed&retmode=json&retmax=num&sort=relevance&term=keyword
```

with no spaces and all on a single line.

So, for example, to find the 20 publications most relevant to "fever", the URL would be:

```
https://eutils.ncbi.nlm.nih.gov/entrez/eutils/esearch.fcgi?
     db=pubmed&retmode=json&retmax=20&sort=relevance&term=fever
```

with no spaces and all on a single line.

When you read the contents of this web page, it is in the JSON format. You can turn this into a dictionary easily using the module `json`. The method `json.loads(`*`contents`*`)`, where *contents* is the contents of the web page, returns a dictionary with one entry, the key of which is "esearchresults". The associated value is another dictionary. The part you want is a list of the publication numbers. The key is "idlist" and the value is a list of the numbers.

You are to print the numbers of that list on a single line, with commas between them (no spaces). So, for the above, your output would look like this:

```
5822579,26772198,20660880,9208885,24176478,27209095,8698996,10913413,24176479,16895496,
24176472,2200377,29940346,8272282,7567198,7432877,26514056,3056881,23160839,19578318
```

but all on a single line. Note your numbers might differ from these because more relevant publications may be found.

Please prompt the user for the keyword to search for.

To turn in: Please call your program *pubmed.py* and submit it to Canvas

A Problem You May Encounter, and Its Solution

If you get the following error (it will be on two lines):

```
We failed to reach a server.
Reason:  [SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed: self signed certificate
      in certificate chain (_ssl.c:1002)
```

that is a problem at the server end that, unfortunately, is causing your connection to PubMed to fail. To solve it, import the module "ssl" and then put the following anywhere *before* you go to the web site:

```
try:
    _create_unverified_https_context = ssl._create_unverified_context
except AttributeError:
    # Legacy Python that doesn't verify HTTPS certificates by default
    pass
else:
    # Handle target environment that doesn't support HTTPS verification
    ssl._create_default_https_context = _create_unverified_https_context
```

In case you want to know what's going on (and if you don't, skip this part), when you connect to a site using "https:", the server sends a *certificate* to your browser to verify that the client (your browser or this program) went to the right place. If this check fails, or the certificate cannot be validated for some reason, it will be rejected by your client. If the client is a browser, you usually get a message that says something like “Bad certificate” or “Unable to verify certificate”. In this program, you will get the error message above. The above Python lines tell your program to ignore this error.

Here’s what the above means. “ssl” is a module that handles secure connections; you can tell these by the “https:” in the URL. By default, it analyzes the certificate, and does the rejection as described above. The attribute “_create_unverified_context” says that the ssl module is to ignore the certificate (the “unverified” part). The `except` part is for versions of the ssl module that do not check certificate validity, and says to ignore that the attribute doesn’t exist. If it does exist, then the `else` part sets the module to ignore any errors with the certificate.

In more detail, the ssl module checks certificate validity by default. If the attribute “_create_unverified_context” does not exist, the ssl module is an old module that does not check certificate validity; that the attribute does not exist causes an AttributeError, and in this case we don’t need to do anything. If it does exist, the default context for the new instance of ssl is set to that attribute, meaning the ssl module will not check certificate validity.