Jaydeep Dey 20BCE1419

Question 1

In [ ]:

```
import requests
from bs4 import BeautifulSoup
import re
```

In [ ]:

```
root_URL = "http://www.vit.ac.in"
search_word = "research"
```

In [ ]:

```
response = requests.get(root_URL)
print("Status of the response : ", response.status_code)
```

```
Status of the response :  200
```

In [ ]:

```
root_page = BeautifulSoup(response.content, 'html.parser')
```

In [ ]:

```
# Retrieve all the links to the sub-pages by retrieving all the `` tags

anchor_tags = root_page.find_all('a')

result = []

# Check if the word "admission" is present in each page, and if so then save its URL
for anchor_tag in anchor_tags :
    link = anchor_tag['href']
    if re.search(search_word, link, re.IGNORECASE) :
        result.append(link)
```

In [ ]:

```
print("The links in the root URL page which contains the word 'research' are :")
for url in result :
    print("\t", url)
```

```
The links in the root URL page which contains the word 'research' are :
         https://vit.ac.in/admissions/research
         https://vit.ac.in/research
         https://vit.ac.in/research
         https://vit.ac.in/research/academic
         https://vit.ac.in/research/sponsored-research
         https://vit.ac.in/research/centers-list
         https://vit.ac.in/schools-centres-list-research-guides-2022
         3d-printing-play-major-role-mitigating-spread-covid-19-say-researchers-vit
         3d-printing-play-major-role-mitigating-spread-covid-19-say-researchers-vit
         https://vit.ac.in/research
```

Question 2

Find documents that contain the word “admissions” and the word “international” within the URL “Vit.ac.in” using Python.

In [ ]:

```
import requests
from bs4 import BeautifulSoup
import re
```

In [ ]:

```
root_URL = "http://www.vit.ac.in"
search_words = ['admissions', 'international']
```

In [ ]:

```
response = requests.get(root_URL)
print("Status of the response : ", response.status_code)
```

```
Status of the response :  200
```

In [ ]:

```
root_page = BeautifulSoup(response.content, 'html.parser')
```

In [ ]:

```
anchor_tags = root_page.find_all('a')
```

In [ ]:

```
valid_links = []
for anchor_tag in anchor_tags :
    link = anchor_tag['href']
    if link.startswith("http") :
        if link not in valid_links :
            valid_links.append(link)
```

In [ ]:

```
print("The number of documents/pages linked to the current root page is : ", len (va
```

```
The number of documents/pages linked to the current root page is :  166
```

In [ ]:

```
result=[]
failed=[]
```

In [ ]:

```
for link in valid_links :
    try :
        page = requests.get(link).text
    except requests.ConnectionError :
        try :
            page = requests.get(link, verify=False).text
        except :
            failed.append(link)
        continue
    if (re.search(search_words[0], page, re.IGNORECASE)) and (re.search(search_words
        result.append(link)
```

```
c:\Users\jayde\AppData\Local\Programs\Python\Python38\lib\site-packages\urllib3\conn
ectionpool.py:842: InsecureRequestWarning: Unverified HTTPS request is being made. A
dding certificate verification is strongly advised. See: https://urllib3.readthedoc
s.io/en/latest/advanced-usage.html#ssl-warnings
  warnings.warn((
c:\Users\jayde\AppData\Local\Programs\Python\Python38\lib\site-packages\urllib3\conn
ectionpool.py:842: InsecureRequestWarning: Unverified HTTPS request is being made. A
dding certificate verification is strongly advised. See: https://urllib3.readthedoc
s.io/en/latest/advanced-usage.html#ssl-warnings
  warnings.warn((
```

In [ ]:

```python
print("The links in the root URL page which contains the word 'admissions', and 'int
ans = []
for i in range(25):
      ans.append(result[i])
for url in ans :
      print("\t", url)
```

```
The links in the root URL page which contains the word 'admissions', and 'internatio
nal' are :
         https://vitap.ac.in/
         https://vitbhopal.ac.in/
         https://vit.ac.in
         https://vit.ac.in/about-vit
         https://vit.ac.in/about/vision-mission
         https://vit.ac.in/vit-milestones
         https://vit.ac.in/about/leadership
         https://vit.ac.in/governance
         https://vit.ac.in/about/administrative-offices
         https://vit.ac.in/about/infrastructure
         https://vit.ac.in/about/sustainability
         https://vit.ac.in/true-green
         https://vit.ac.in/about/community-outreach
         https://vit.ac.in/about/communityradio
         https://vit.ac.in/all-news-archieved
         https://vit.ac.in/all-events
         https://vit.ac.in/national-institutional-ranking-framework-nirf
         https://vit.ac.in/mhrdugcaicte
         https://vit.ac.in/about/news-letter
         https://vit.ac.in/academics/home
         https://vit.ac.in/programmes-offered-1
         https://vit.ac.in/programmes-offered-2021-22
         https://vit.ac.in/programmes-offered-2020-21
         https://vit.ac.in/schools
         https://vit.ac.in/academics/ffcs
```

In [ ]:

```python
print("The links that we failed to open are : ")
for url in failed :
    print("\t", url)
```

```
The links that we failed to open are :
         http://intranet.vit.ac.in
         http://intranet.vit.ac.in/
```

Question 3

Find documents that contain the word “Programme” but not the word “programming” within the URL “Vit.ac.in” using Python.

In [ ]:

```python
import requests
from bs4 import BeautifulSoup
import re
```

In [ ]:

```python
root_URL = "http://www.vit.ac.in"
search_word_1 = "Programme"
search_word_2 = "Programming"
```

In [ ]:

```python
response = requests.get(root_URL)
print("Status of the response : ", response.status_code)
```

```
Status of the response :  200
```

In [ ]:

```
root_page = BeautifulSoup(response.content, 'html.parser')
```

In [ ]:

```
anchor_tags = root_page.find_all('a')
```

In [ ]:

```
valid_links = []
for anchor_tag in anchor_tags :
    link = anchor_tag['href']
    if link.startswith("http") :
        if link not in valid_links :
            valid_links.append(link)
```

In [ ]:

```
print("The number of documents/pages linked to the current root page is : ", len (va
```

```
The number of documents/pages linked to the current root page is :  166
```

In [ ]:

```
result=[]
failed=[]
```

In [ ]:

```
for link in valid_links :
    try :
        page = requests.get(link).text
    except requests.ConnectionError :
        try :
            page = requests.get(link, verify=False).text
        except :
            failed.append(link)
        continue
    if (re.search(search_word_1, page, re.IGNORECASE)) and (not re.search(search_wor
        result.append(link)
```

```
c:\Users\jayde\AppData\Local\Programs\Python\Python38\lib\site-packages\urllib3\conn
ectionpool.py:842: InsecureRequestWarning: Unverified HTTPS request is being made. A
dding certificate verification is strongly advised. See: https://urllib3.readthedoc
s.io/en/latest/advanced-usage.html#ssl-warnings
  warnings.warn((
```

In [ ]:

```
print("The links in the root URL page which contains the word 'Programme' but not th
ans = []
for i in range(5):
      ans.append(result[i])
for url in ans :
    print("\t", url)
```

```
The links in the root URL page which contains the word 'Programme' but not the word
'programming' are :
         https://vitap.ac.in/
         https://vitbhopal.ac.in/
         https://vit.ac.in
         https://vit.ac.in/about-vit
         https://vit.ac.in/about/vision-mission
```

Question 4

Write a web crawler program which takes as input a url(Educational Website), a search word and maximum number of pages(15-20 Pages) to be searched and returns as output all the web pages it searched till it found the search word on a web page or return failure.

In [ ]:

```
import requests
from bs4 import BeautifulSoup
import re
```

In [ ]:

```
seedURL4 = input("Enter the Input URL:")
searchWord = input("Enter the Search Word: ")
maxPages = int(input("Enter the Max Pages:"))
```

In [ ]:

```
response = requests.get(seedURL4)
print("Status of the response : ", response.status_code)
rootPage=BeautifulSoup(response.content,'html.parser')
```

```
Status of the response :  200
```

In [ ]:

```
atags=rootPage.find_all('a')
validLinks= []
```

In [ ]:

```
for atag in atags:
    try:
        link=atag['href']
        if link.startswith("http") :
            if link not in validLinks :
                validLinks.append(link)
    except:
        pass
print("Total Number of Documents is {}".format(len(validLinks)))
```

```
Total Number of Documents is 124
```

In [ ]:

```
final= []
foundPages=0
failed= []
pages=0
```

In [ ]:

```
for link in validLinks :
    if(pages==maxPages):
        break
    try :
        page = requests.get(link).text
    except requests.ConnectionError :
        try :
            page = requests.get(link, verify=False).text
        except :
            failed.append(link)
        continue
    if (re.search(searchWord, page, re.IGNORECASE)):
        final.append(link)
        foundPages+=1
    pages+=1
if(foundPages==0):
    print("Failure")
else:
    print("The Documents that Contain the Word {} is ".format(searchWord))
    for i in final:
        print("\t",i)
```

```
c:\Users\jayde\AppData\Local\Programs\Python\Python38\lib\site-packages\urllib3\conn
ectionpool.py:842: InsecureRequestWarning: Unverified HTTPS request is being made. A
dding certificate verification is strongly advised. See: https://urllib3.readthedoc
s.io/en/latest/advanced-usage.html#ssl-warnings
  warnings.warn((
The Documents that Contain the Word research is
         http://www.vit.ac.in/
         http://vitbhopal.ac.in/
         https://vitap.ac.in
         https://vitap.ac.in/admission/overview/
         https://vitap.ac.in/btech/
         https://vitap.ac.in/vit-ap-school-of-business/
         https://vitap.ac.in/school-of-law/
         https://vitap.ac.in/m-a/
         https://vitap.ac.in/bscmsc/
         https://vitap.ac.in/m-sc/
```

Question 5

Write a Python program to read the given website and extract the phone numbers and emails and contact addresses from Chennai, Amaravathi, Bhopal vit website.

In [ ]:

```
from bs4 import BeautifulSoup
import requests
```

In [ ]:

```
seedUrl = ["https://vit.ac.in/","https://chennai.vit.ac.in/", "https://vitap.ac.in/"

f = open("link.txt", "w")
```

In [ ]:

```
phonePattern = re.compile(r'[7-9][0-9]{9}')
emailpattern= re.compile(r'\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Z|a-z]{2,}\b')

for url in seedUrl:
    response = requests.get(url, verify=False)
    phone = re.findall(phonePattern, response.text)
    email = re.findall(emailpattern, response.text)
    f.write(' '.join(email))
    f.write(' '.join(phone))
f.close()
f2 = open('link.txt', 'r')
print(f2.read())
```

```
c:\Users\jayde\AppData\Local\Programs\Python\Python38\lib\site-packages\urllib3\conn
ectionpool.py:842: InsecureRequestWarning: Unverified HTTPS request is being made. A
dding certificate verification is strongly advised. See: https://urllib3.readthedoc
s.io/en/latest/advanced-usage.html#ssl-warnings
  warnings.warn((
c:\Users\jayde\AppData\Local\Programs\Python\Python38\lib\site-packages\urllib3\conn
ectionpool.py:842: InsecureRequestWarning: Unverified HTTPS request is being made. A
dding certificate verification is strongly advised. See: https://urllib3.readthedoc
s.io/en/latest/advanced-usage.html#ssl-warnings
  warnings.warn((
c:\Users\jayde\AppData\Local\Programs\Python\Python38\lib\site-packages\urllib3\conn
ectionpool.py:842: InsecureRequestWarning: Unverified HTTPS request is being made. A
dding certificate verification is strongly advised. See: https://urllib3.readthedoc
s.io/en/latest/advanced-usage.html#ssl-warnings
  warnings.warn((
c:\Users\jayde\AppData\Local\Programs\Python\Python38\lib\site-packages\urllib3\conn
ectionpool.py:842: InsecureRequestWarning: Unverified HTTPS request is being made. A
dding certificate verification is strongly advised. See: https://urllib3.readthedoc
s.io/en/latest/advanced-usage.html#ssl-warnings
  warnings.warn((
9398902106cw.cc@vit.ac.in wlh.cc@vit.ac.in transport.cc@vit.ac.in cw.cc@vit.ac.in wl
h.cc@vit.ac.in transport.cc@vit.ac.in admin.chennai@vit.ac.in8272122876 8272215041 8
272239968 8272270359 8272298574 8659071608 8588684987 8679772278 8634261762 86342700
77 9038225429 7121194580 8634261762 8588708437 8634261762 8588708437 8201114257 8201
114257 8588708437 8588708437 8659971350 9315459946 8679772278 7358782569 8272122876
8634261762 9038225429 8634270077 7121194580 7358782569 8634261762 8272215041 8588708
437 8272239968 8634261762 8272270359 8588708437 8272298574 8201114257 8201114257 858
8708437 8659071608 8588708437 8659971350 9315459946 85886849877868934148 7868934148
7868934148 7797590012 9520281387info@vitbhopal.ac.in placement@vitbhopal.ac.in admis
sions@vitbhopal.ac.in wardenmh@vitbhopal.ac.in wardenlh@vitbhopal.ac.in9662422233 81
94455437 8262477098 8876659932 7795006127 8612877226 7709999738 7986318372 884591017
1 8876348337 8945379300 8945479108 8612870136 8845918969 8845910863 8845911379 88459
11923 8194455437 8262477098 8876348337 8945379300 8945479108 8876659932 7795006127 8
612877226 8612870136 7709999738 8061303498 7035880534 8864788979 8061303498 70358805
34 8864788979 7035880534 8864788979 7035880534 8864788979 7035880534 8864788979 7035
880534 8864788979 8061303498 8264704521 7465352127 7174252163 8551857321 8264704521
7465352127 7174252163 7465352127 7174252163 7465352127 7174252163 7465352127 7174252
163 7465352127 7174252163 7979272149 9344318998 8728174513 7979272149 9344318998 934
4318998 9344318998 9344318998 9344318998 7906040746 8286208450 7317899495 9568028126
7906040746 8286208450 7317899495 8286208450 7317899495 8286208450 7317899495 8286208
450 7317899495 8286208450 7317899495 8150108223 8318240920 9602035035 8797830589 815
0108223 8318240920 9602035035 8318240920 9602035035 8318240920 9602035035 8318240920
9602035035 8318240920 9602035035 7977097410 9741738558 8399911470 7662683301 7977097
410 9741738558 8399911470 9741738558 8399911470 9741738558 8399911470 9741738558 839
9911470 9741738558 8399911470 8324916849 9672133224 8482824318 8324916849 9672133224
8482824318 9672133224 8482824318 9672133224 8482824318 9672133224 8482824318 9672133
224 8482824318 8324916849 8113170000 7729692478 7729692478 7397009434 9618935664 926
0750302 7217597803 7075121594 9065360540 9275123713 8756951133 8955924605 9158682827
9655273119 8113170000 7729692478 7729692478 7729692478 7729692478 7729692478 7939801
444 8213782599 7254431471 8213782599 7254431471 8503397689 7710044413 7859852193 752
8554828 8246052324 7107344378 9287253478 9906576570 8377194160 8351581336 7089927560
8180286903 8443864678 8797978927 7939801444 8213782599 7254431471 8213782599 7254431
471 8213782599 7254431471 8213782599 7254431471 8213782599 7254431471 7979757306 811
0484234 7831648607 7979757306 8110484234 7831648607 8110484234 7831648607 8110484234
7831648607 8110484234 7831648607 8110484234 7831648607 7979757306 7997123712 8061489
449 8187953360 7997123712 8061489449 8187953360 8061489449 8187953360 8061489449 818
7953360 8061489449 8187953360 8061489449 8187953360 7997123712 7953510457 8358086840
7958327780 7953510457 8358086840 7958327780 8358086840 7958327780 8358086840 7958327
780 8358086840 7958327780 8358086840 7958327780 7953510457 7926112081 8941056685 756
3129879 7926112081 8941056685 7563129879 8941056685 7563129879 8941056685 7563129879
8941056685 7563129879 8941056685 7563129879 7926112081 7893933214 7998754603 9141007
060 7893933214 7998754603 9141007060 7998754603 9141007060 7998754603 9141007060 799
8754603 9141007060 7998754603 9141007060 7893933214 7888279366 7078686666 9281519859
7888279366 7078686666 9281519859 7078686666 9281519859 7078686666 9281519859 7078686
666 9281519859 7078686666 9281519859 7888279366 7892372833 7097665995 9446229641 789
2372833 7097665995 9446229641 7097665995 9446229641 7097665995 9446229641 7097665995
9446229641 7097665995 9446229641 7892372833 7968996840 7496227456 8514161077 7968996
840 7496227456 8514161077 7496227456 8514161077 7496227456 8514161077 7496227456 851
4161077 7496227456 8514161077 7968996840 7974585219 9986160257 7917539790 7974585219
9986160257 7917539790 9986160257 7917539790 9986160257 7917539790 9986160257 7917539
790 9986160257 7917539790 7974585219 8312719179 8851818198 9034194288 8851818198 903
4194288 9216018669 9344396816 8998725771 7553521655 8435424529 8213626783 8312719179
8851818198 9034194288 8851818198 9034194288 8851818198 9034194288 8851818198 9034194
288 8851818198 9034194288 7970312688 7884614597 7449419967 7970312688 7884614597 744
9419967 7884614597 7449419967 7884614597 7449419967 7884614597 7449419967 7884614597
7449419967 7970312688 7970312688 7884614597 7449419967 8312719179 8851818198 9034194
288 7968996840 7496227456 8514161077 7974585219 9986160257 7917539790 7892372833 709
7665995 9446229641 7888279366 7078686666 9281519859 7926112081 8941056685 7563129879
7893933214 7998754603 9141007060 7953510457 8358086840 7958327780 7979757306 8110484
234 7831648607 7997123712 8061489449 8187953360 8324916849 9672133224 8482824318 811
3170000 7729692478 7939801444 8213782599 7254431471 8264704521 7465352127 7174252163
7979272149 9344318998 7906040746 8286208450 7317899495 8150108223 8318240920 9602035
035 7977097410 9741738558 8399911470 8061303498 7035880534 8864788979 7986318372 884
5910171 8845918969 8845910863 8845911379 8845911923 9212651252
```