Welcome to the 2nd Assignment of this unit. This assignment will test your research and programming skills. You can use all the available online resources to complete this assignment.

Question 1

This question consists of multiple CSV files (In the Zipped Folder) with 'large texts' in one of the columns in each file. Your job is to use the open-source NLP (Natural Language Processing) libraries and perform various tasks.

Task 1:

Extract the 'text' in all the CSV files and store them into a single '.txt file'.

Task 2: Research

Install the libraries(SpaCy – scispaCy – 'en_core_sci_sm'/'en_ner_bc5cdr_md').

Install the libraries (Transformers (Hugging Face) - and any bio-medical model (BioBert) that can detect drugs, diseases, etc from the text).

Task 3: Programming and Research

3.1:

Using any in-built library present in Python, count the occurrences of the words in the text (.txt) and give the 'Top 30' most common words.

And store the 'Top 30' common words and their counts into a CSV file.

3.2:

Using the 'Auto Tokenizer' function in the 'Transformers' library, write a 'function' to count unique tokens in the text (.txt) and give the 'Top 30' words.

Task 4: Named-Entity Recognition (NER)

Extract the 'diseases', and 'drugs' entities in the '.txt file' separately using 'en_core_sci_sm'/'en_ner_bc5cdr_md' and biobert. And compare the differences between the two models (Example: Total entities detected by both of them, what's the difference, check for most common words, and check the difference.)

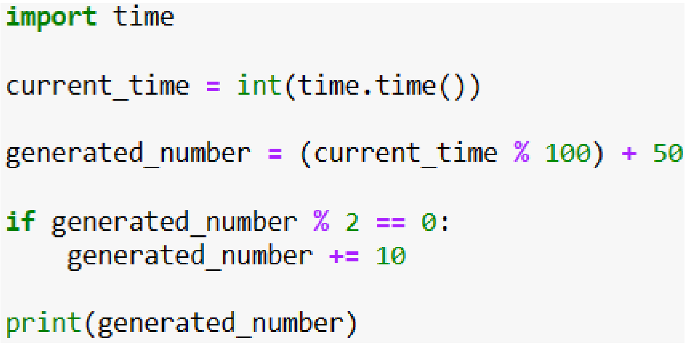
Question 2

Here's an adventurous story intertwined with Python programming questions that involve nested for loops, conditional statements, string manipulations, and more.

The Quest for the Hidden Treasure:

Deep within the mystical lands of Pythoria lay the fabled Temple of Codes, rumored to house a treasure of knowledge guarded by enigmatic puzzles. The path is challenging, and only those who can do the coding will unravel the final word, leading to the treasure.

Chapter 1: The Gatekeeper



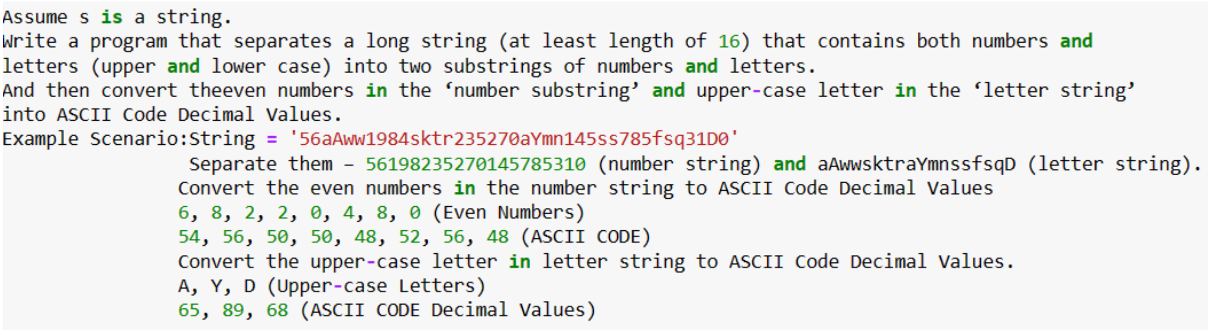
```
import time

current_time = int(time.time())

generated_number = (current_time % 100) + 50

if generated_number % 2 == 0:
    generated_number += 10

print(generated_number)
```

The above algorithm generates a number (n). You should use this number to change the pixels (r,g,b) in the provided image (Chapter1.png) by adding the original pixel values (r,g,b) with the generated number (Example: (r+n, g+n, b+n)).

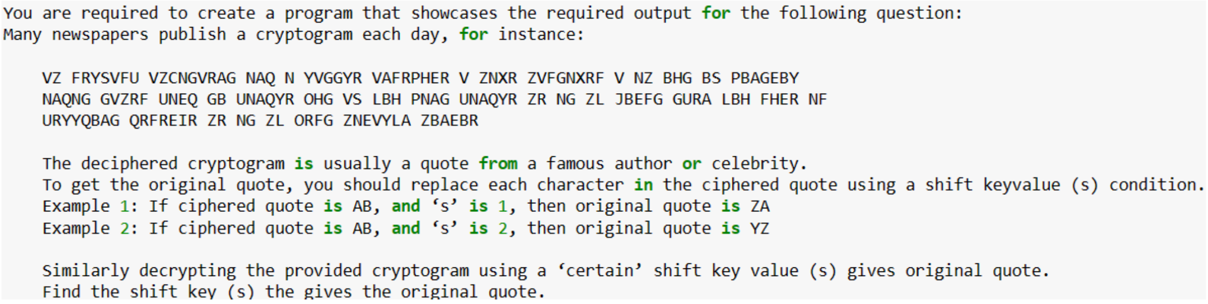
Generate a new image with the converted pixels (upload it as 'chapter1out.png').

Finally, add all the red (r) pixel values in the new_image and provide the sum as output to move to the next chapter.

Chapter 2: The Chamber of Strings



```
Assume s is a string.
Write a program that separates a long string (at least length of 16) that contains both numbers and
letters (upper and lower case) into two substrings of numbers and letters.
And then convert theeven numbers in the ‘number substring’ and upper-case letter in the ‘letter string’
into ASCII Code Decimal Values.
Example Scenario:String = '56aAww1984sktr235270aYmn145ss785fsq31D0'
                 Separate them – 56198235270145785310 (number string) and aAwwsktraYmnssfsqD (letter string).
                Convert the even numbers in the number string to ASCII Code Decimal Values
                6, 8, 2, 2, 0, 4, 8, 0 (Even Numbers)
                54, 56, 50, 50, 48, 52, 56, 48 (ASCII CODE)
                Convert the upper-case letter in letter string to ASCII Code Decimal Values.
                A, Y, D (Upper-case Letters)
                65, 89, 68 (ASCII CODE Decimal Values)
```

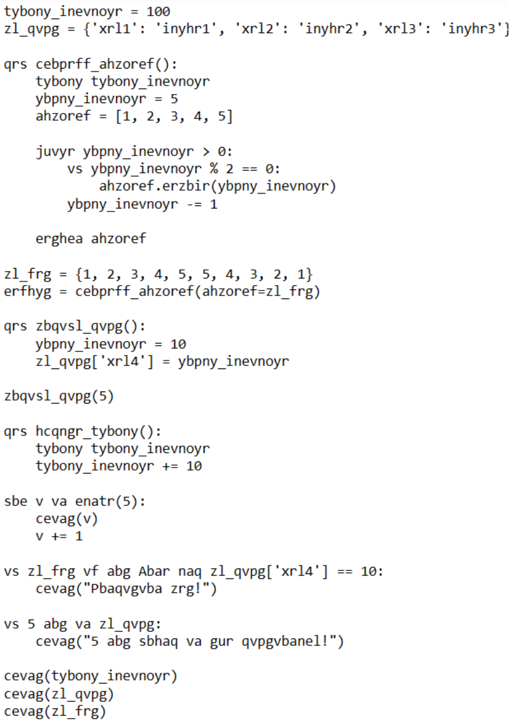


```
You are required to create a program that showcases the required output for the following question:
Many newspapers publish a cryptogram each day, for instance:

    VZ FRYSVFU VZCNGVRAG NAQ N YVGGYR VAFRPHER V ZNXR ZVFGNXRF V NZ BHG BS PBAGEBY
    NAQNG GVZRF UNEQ GB UNAQYR OHG VS LBH PNAG UNAQYR ZR NG ZL JBEFG GURA LBH FHER NF
    URYYQBAG QRFREIR ZR NG ZL ORFG ZNEVYLA ZBAEBR

    The deciphered cryptogram is usually a quote from a famous author or celebrity.
    To get the original quote, you should replace each character in the ciphered quote using a shift keyvalue (s) condition.
    Example 1: If ciphered quote is AB, and ‘s’ is 1, then original quote is ZA
    Example 2: If ciphered quote is AB, and ‘s’ is 2, then original quote is YZ

    Similarly decrypting the provided cryptogram using a ‘certain’ shift key value (s) gives original quote.
    Find the shift key (s) the gives the original quote.
```

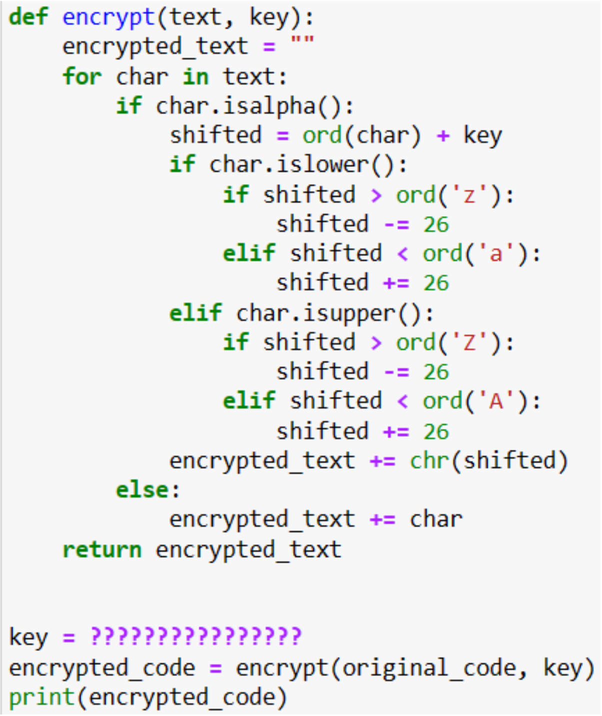
Question 3

Fixing the error-prone codes.

Below is the code (left) that is encrypted using a number. Once you decrypt the below code, it reveals the original code with many errors. Please fix them and explain them using comments (#).



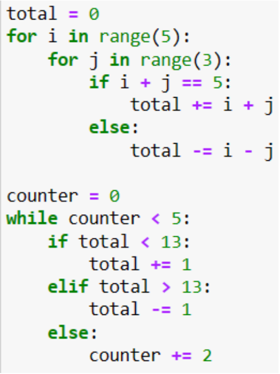
```
tybony_inevnoyr = 100
zl_qvpg = {'xrl1': 'inyhr1', 'xrl2': 'inyhr2', 'xrl3': 'inyhr3'}

qrs cebprff_ahzoref():
    tybony tybony_inevnoyr
    ybpny_inevnoyr = 5
    ahzoref = [1, 2, 3, 4, 5]

    juvyr ybpny_inevnoyr > 0:
        vs ybpny_inevnoyr % 2 == 0:
            ahzoref.erzbir(ybpny_inevnoyr)
        ybpny_inevnoyr -= 1

    erghea ahzoref

zl_frg = {1, 2, 3, 4, 5, 5, 4, 3, 2, 1}
erfhyg = cebprff_ahzoref(ahzoref=zl_frg)

qrs zbqvsl_qvpg():
    ybpny_inevnoyr = 10
    zl_qvpg['xrl4'] = ybpny_inevnoyr

zbqvsl_qvpg(5)

qrs hcqngr_tybony():
    tybony tybony_inevnoyr
    tybony_inevnoyr += 10

sbe v va enatr(5):
    cevag(v)
    v += 1

vs zl_frg vf abg Abar naq zl_qvpg['xrl4'] == 10:
    cevag("Pbaqvgvba zrg!")

vs 5 abg va zl_qvpg:
    cevag("5 abg sbhaq va gur qvpgvbanel!")

cevag(tybony_inevnoyr)
cevag(zl_qvpg)
cevag(zl_frg)
```



```
def encrypt(text, key):
    encrypted_text = ""
    for char in text:
        if char.isalpha():
            shifted = ord(char) + key
            if char.islower():
                if shifted > ord('z'):
                    shifted -= 26
                elif shifted < ord('a'):
                    shifted += 26
            elif char.isupper():
                if shifted > ord('Z'):
                    shifted -= 26
                elif shifted < ord('A'):
                    shifted += 26
            encrypted_text += chr(shifted)
        else:
            encrypted_text += char
    return encrypted_text


key = ????????????????
encrypted_code = encrypt(original_code, key)
print(encrypted_code)
```

To decrypt the above code, first, you need to understand how it is encrypted (above right image).

1. Fixing the next code will reveal the key.
2. Write the decryption function to decrypt the ‘encrypted code’ to the original code.
3. Correct the errors and provide the comments.
4. Should show everything in your program file.



```
total = 0
for i in range(5):
    for j in range(3):
        if i + j == 5:
            total += i + j
        else:
            total -= i - j

counter = 0
while counter < 5:
    if total < 13:
        total += 1
    elif total > 13:
        total -= 1
    else:
        counter += 2
```

Question 4

Welcome to the final task of this assignment. You are required to create a GitHub repository and add all your group mates to it (make sure to keep it public, not private). You should do this before you start the assignment.

All the answers and contributions should be recorded in GitHub till you submit the assignment.

Submission Guidelines

- **Zip all the programming files and outputs and upload them to Learline.**
- **Include your GitHub Repository link in the programming files that you submit.**