Lab 3.2 : (Deadline Friday 4 February 23:59)

- Upload your code to Einstein to have it verified.

More list comprehensions

- Write a program called `wordcomps_032.py` that reads words from `stdin` (one word per line) and stores them all in a list.
- Using *list comprehensions* and ignoring differences in case, have your program print the following:

  1. The shortest word that contains all vowels (‘aeiou’).
  2. A count of all the words that end in ‘iary’.
  3. A list of the words that contain most e’s.
- Your program should produce the following output when run against dictionary05.txt:

```
$ python3 wordcomps_032.py < dictionary05.txt
Shortest word containing all vowels: Sequoia
Words ending in iary: 14
Words with most e's: ['dereference', 'teleconference']
```

Reverse words

- Write a program called `reversecomp_032.py` that reads words from `stdin` (one word per line) and stores them all in a list.
- Making use of a *list comprehension* and ignoring differences in case, have the program print a list of all words that are at least five characters long and whose reverse also occurs in the list.
- Your program should produce the following output when run against dictionary05.txt:

```
$ python3 reversecomp_032.py < dictionary05.txt
['Ababa', 'civic', 'Damon', 'Hannah', 'lager', 'leper', 'level', 'lever', 'madam', 'mi
```

- Note that palindromes will appear in the program output since by definition their reverse is in the list of words.
- Note there is a timeout on the program checker that will halt your program if it does not *efficiently* produce its answer.
- You might use *binary search* (as covered in CA116) over the list of sorted words in order to efficiently solve this problem. Here is some code you might use:

```
# Binary search (adapted from CA116)
def binsearch(query, sorted_list):

    '''Return True if query in sorted_list else False'''

    low = 0
    high = len(sorted_list) - 1
    while low <= high:
        mid = (low + high) // 2

        # print(f'{low} {mid} {high}')

        if sorted_list[mid] < query:
            # Search RHS
            low = mid + 1

        elif sorted_list[mid] > query:
            # Search LHS
            high = mid - 1

        else:
            # Found it
            return True

    # Not found
    return False
```

- Using a dictionary or a set to solve the exercise is not permitted.

Censor

- Write a program called `censor_032.py` that reads a list of censored strings from a file supplied on the command line.
- The program should then read the text supplied on `stdin` and output the same but with each censored string replaced by a string of ampersands of the same length. For example:

```
$ cat censor_input_00_032.txt
low
rose
smell
he
```

```
$ cat censor_stdin_00_032.txt
Sonnet 98
by William Shakespeare

From you have I been absent in the spring,
When proud-pied April dress’d in all his trim
Hath put a spirit of youth in every thing,
That heavy Saturn laugh’d and leap’d with him.
Yet nor the lays of birds nor the sweet smell
Of different flowers in odour and in hue
Could make me any summer’s story tell,
Or from their proud lap pluck them where they grew;
Nor did I wonder at the lily’s white,
Nor praise the deep vermilion in the rose;
They were but sweet, but figures of delight,
Drawn after you, you pattern of all those.
Yet seem’d it winter still, and, you away,
As with your shadow I with these did play.
```

```
$ python3 censor_032.py censor_input_00_032.txt < censor_stdin_00_032.txt
Sonnet 98
by William Shakespeare
```

```
From you have I been absent in t@@ spring,
W@@n proud-pied April dress’d in all his trim
Hath put a spirit of youth in every thing,
That @@avy Saturn laugh’d and leap’d with him.
Yet nor t@@ lays of birds nor t@@ sweet @@@@@
Of different f@@@ers in odour and in hue
Could make me any summer’s story tell,
Or from t@@ir proud lap pluck t@@m w@@re t@@y grew;
Nor did I wonder at t@@ lily’s white,
Nor praise t@@ deep vermilion in t@@ @@@@;
T@@y were but sweet, but figures of delight,
Drawn after you, you pattern of all those.
Yet seem’d it winter still, and, you away,
As with your shadow I with t@@se did play.
```

- Case should be ignored when looking for censored strings. Original case should be retained however in the output.
- Replace censored strings in the order they are listed i.e. in the example above all instances of *rose* should be replaced before all instances of *smell*.