CHE 120 Assignment 4

Anna Kelley

21201579

November 29, 2025

Department of Chemical Engineering

Table of Contents

Table of Figures

1. Question 1

1.1. Approach

To begin, I know that I will need to code one function and a main program that executes this function. Inside the move list function, I will check input, then create a new list based on how many loops the code will run which is determined by variable n.

1.2. Result

The first result is from a function call where the input matches the examples given on the project outline

```
In [25]: %runfile 'C:/Users/annak/Downloads/che 120/moving.py' --wdir
[4, 5, 1, 2, 3]
[1, 2, 3, 4, 5]
[4, 5, 1, 2, 3]
[4, 5, 'a', 'b', 'hello']
```

Figure 1.1 Program run with examples from problem

The second result is from a function call where the input is not a list so it is invalid.

```
In [27]: %runfile 'C:/Users/annak/Downloads/che 120/
moving.py' --wdir
Invalid input
None
```

Figure 1.1 Program run with invalid input

1.3. Discussion

This result was achieved by moving the items in the list forward one index repeatedly. By making a new list which is a copy of the function list, this is possible. For the loop, the last element is moved to the first index of the list. Then it repeats that until the new list is made.

2. Question 2

2.1. Approach

To begin, I know that I will need to code one function called create collection and a main program that executes this function. Inside the create collection function, I will check input, then return a new collection based on what is in the function arguments

2.2. Result

The result is shown using all the examples in the problem statement

```
In [30]: %runfile 'C:/Users/annak/Downloads/che 120/
collection.py' --wdir
['h', 'e', 'l', 'l', 'o']
('h', 'e', 'l', 'l', 'o')
{'o', 'l', 'h', 'e'}
{0: 'h', 1: 'e', 2: 'l', 3: 'l', 4: 'o'}
dictionary is not a valid type string
None
```

Figure 2.1 Output image for test results

2.3. Discussion

The result of this function is an output of whatever string in the argument as a collection specified in the other output. Each collection (tuple, set, dict, list) have different initialization and syntax, so separate if statements are made for each. I went through every statement before list, because that is the default value so can just use an else statement if there is no argument.

3. Question 3

3.1. Approach

For this question, I have to open a file, read the file, then split the file into each word and remove any invalid words. I will do all of this in the main script. Once the text file is separated into words, create a new dictionary based on the frequency of each word appearing. This dictionary will then be sorted and the first 5 entries (words with highest frequency) will be outputted.

.

3.2. Results

Result of the Charles-Dicken's Oliver Twist file, the top 5 most frequent words have been printed

```
In [31]: %runfile 'C:/Users/annak/Downloads/che 120/
Question3Assignment4.py' --wdir
The 5 words with the highest frequency in file 'CharlesDickens-
OliverTwist.txt' are:
1: 'the' appears 9771 times
2: 'and' appears 5425 times
3: 'of' appears 3981 times
4: 'to' appears 3948 times
5: 'a' appears 3785 times
```

Figure 3.1 Charles Dickens file example

3.3. Discussion

For this problem, I first opened the text file and put all the text into a string. From there I was able to split the string into words using the .split() method. Then, for each word in the words string, a new word was created as a string. Each word is checked to have only an alphabetic characters, and if they don't, those characters are replaced. Additionally, I set each word to lowercase. From there, I created a new list with all of these words. Then, for each word in the list, check to see if the word is already in the dictionary. If it isn't then create a new entry in the dictionary with the word. If it is already there, increase the value of the word. Then, once this is done there will be a list with the keys of each word and the values of how many times that word appears. After this dictionary is sorted, I used the sorted function to sort the dictionary based on its keys. Then, I used this sorted

keys dictionary to create the sorted dictionary with the right values. Then I just went through the first 5 entries of the list and outputted them.