3a. Provide a written response that does all three of the following:

i. Describes the overall purpose of the program

The purpose of this program is to generate a variety of strong, secure passwords for people to choose from.

ii. Describes what functionality of the program is demonstrated in the video

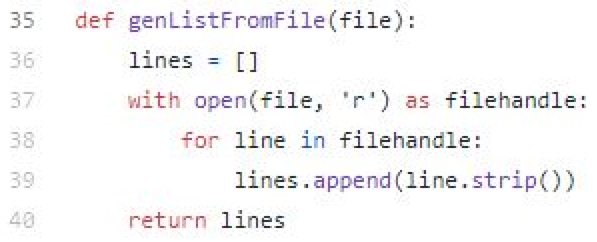
My program functions by continually printing random passwords to the terminal. After each print the user can type any key to continue, or x if they are satisfied with the password and would like to stop the program.

iii. Describes the input and output of the program demonstrated in the video

The input to the program is two text files. A string(text) is also entered as input through the terminal. The output is a string(text) printed to the terminal. The interface is a keyboard and computer monitor.

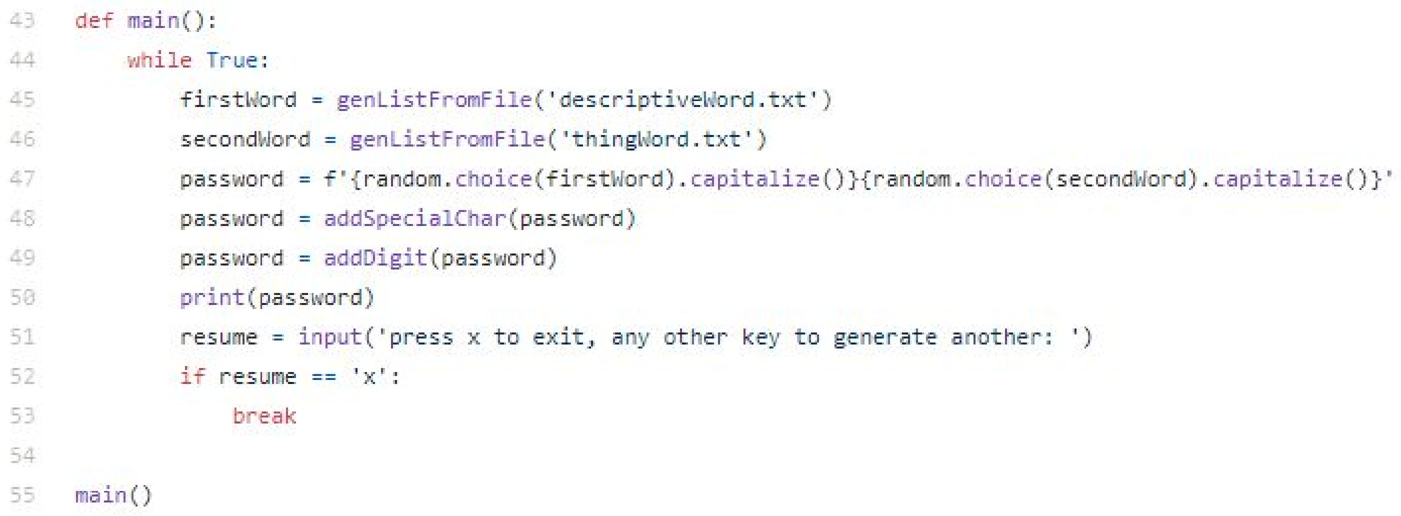
3b. Capture and paste two program code segments you developed during the administration of this task that contain a list (or other collection type) being used to manage complexity in your program.

i. The first program code segment must show how data have been stored in the list.



```python
def genListFromFile(file):
    lines = []
    with open(file, 'r') as filehandle:
        for line in filehandle:
            lines.append(line.strip())
    return lines
```

ii. The second program code segment must show the data in the same list being used, such as creating new data from the existing data or accessing multiple elements in the list, as part of fulfilling the program's purpose.



```python
def main():
    while True:
        firstWord = genListFromFile('descriptiveWord.txt')
        secondWord = genListFromFile('thingWord.txt')
        password = f'{random.choice(firstWord).capitalize()}{random.choice(secondWord).capitalize()}'
        password = addSpecialChar(password)
        password = addDigit(password)
        print(password)
        resume = input('press x to exit, any other key to generate another: ')
        if resume == 'x':
            break

main()
```

iii. Identifies the name of the list being used in the response

firstWord

iv. Describes what the data contained in the list represent in your program

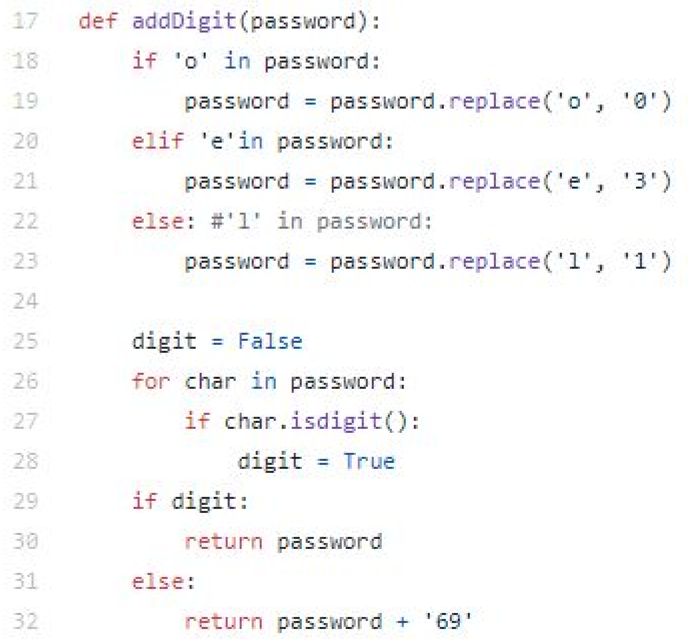
Each element of firstWord is a unique word(string) from the English language that is generally either an adjective or adverb. They represent strings that will be randomly selected and concatenated with other strings to create a composite password.

v. Explains how the selected list manages complexity in your program code by explaining why your program code could not be written, or how it would be written differently, if you did not use the list.

My list firstWord manages complexity because it is dynamic, it can be as long or short as necessary based on how many words are in the text document it is populated from. To recreate this using individual strings; I would have to know ahead of time how many words I wanted and couldn't easily accommodate changes in the data size. The list enables me to choose a random word, which I can't recreate with individual variables. The list makes it easier to maintain because the data is separated from the program. I can edit the text document to alter the data thus reducing the risk of editing the source code by mistake.

3c. Capture and paste two program code segments you developed during the administration of this task that contain a student-developed procedure that implements an algorithm used in your program and a call to that procedure.

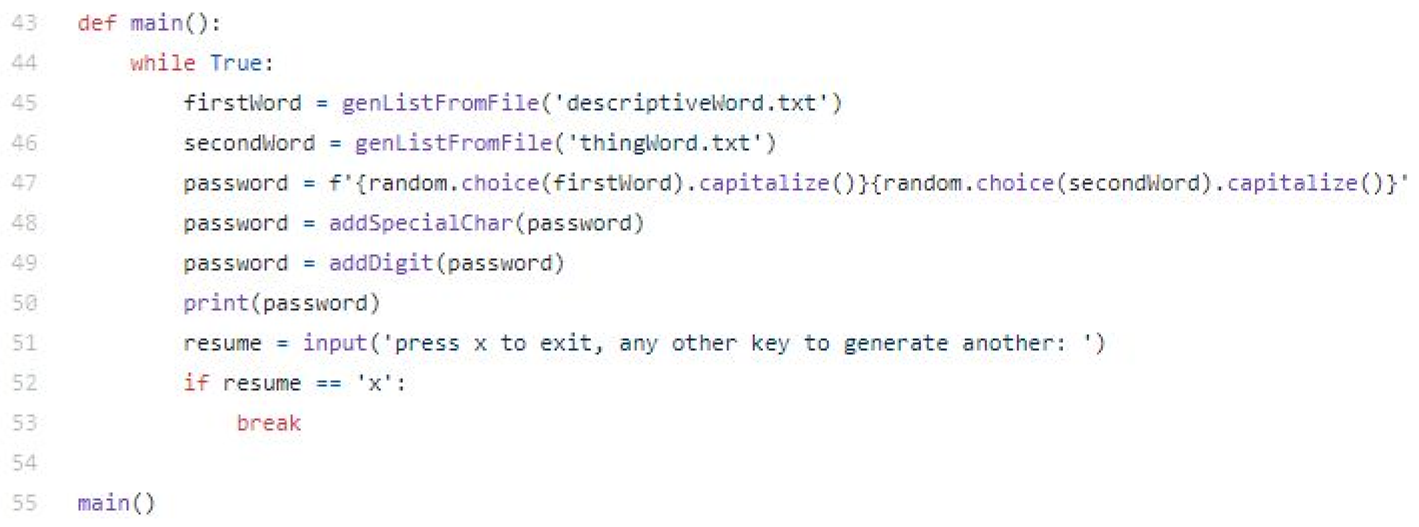
i. The first program code segment must be a student-developed procedure that:
- Defines the procedure's name and return type(if necessary)
- Contains and uses one or more parameters that have an effect on the functionality of the procedure
- Implements an algorithm that includes sequencing, selection, and iteration



```python
def addDigit(password):
    if 'o' in password:
        password = password.replace('o', '0')
    elif 'e'in password:
        password = password.replace('e', '3')
    else: #'l' in password:
        password = password.replace('l', '1')

    digit = False
    for char in password:
        if char.isdigit():
            digit = True
    if digit:
        return password
    else:
        return password + '69'
```

ii. The second program code segment must show where your student-developed procedure is being called in your program



```python
def main():
    while True:
        firstWord = genListFromFile('descriptiveWord.txt')
        secondWord = genListFromFile('thingWord.txt')
        password = f'{random.choice(firstWord).capitalize()}{random.choice(secondWord).capitalize()}'
        password = addSpecialChar(password)
        password = addDigit(password)
        print(password)
        resume = input('press x to exit, any other key to generate another: ')
        if resume == 'x':
            break

main()
```

iii. Describes in general what the identified procedure does and how it contributes the overall functionality of the program

The procedure addDigit contributes to the program functionality by ensuring that each generated password contains a digit. This ensures a more secure password is generated.

iv. Explain in detailed steps how the algorithm implemented in the identified procedure works. Your explanation must be detailed enough for someone else to recreate it.

The algorithm uses selections to determine whether the character ‘o’, ‘e’, or ‘l’ is in the passed argument password. Using String boolean methods; the selection checks and either replaces all ‘o’s with ‘0’s, or all ‘e’s with ‘3’s, or all ‘l’s with ‘1’s.

Next a boolean is used to determine if the password has digits. The password is looped with the loop variable ‘char’ copying each character of password. During each iteration ‘char’ is tested using the string method isdigit(). If the result is True the boolean is set to True.

After the loop a selection evaluates the boolean, if it is True then the updated password is returned. If the boolean is still False, the password is concatenated with preset digits and returned instead so that the final password has digits.

3d. Provide a written response that does all three of the following:

i. Describes two calls to the procedure identified in written response 3c. Each call must pass a different argument(s) that causes a different segment of code in the algorithm to execute.

First Call
The program loops as long as the user desires. During each loop the procedure is called on line 49.

Second Call
The program loops as long as the user desires. During each loop the procedure is called on line 49.

ii. Describes what conditions(s) is being tested by each call to the procedure

Condition(s) tested by the first call:
When the procedure is called it tests whether the argument has the characters ‘o’, ‘e’, or ‘l’ in it.

Condition(s) tested by the first call:
When the procedure is called it tests whether the argument has the characters ‘o’, ‘e’, or ‘l’ in it.

iii. Identifies the result of each call

Result of the first call:

In this example ‘H@ppyPl@net’ is passed to addDigit as the argument. The letter ‘e’ would be identified in line 20 causing it to be converted to the number ‘3’. This would cause the outcome of the boolean loop on lines 25-28 to evaluate to True. This would cause line 30 to execute thus returning the current value of password as ‘H@ppyPl@n3t’.

Result of the second call:

In this example ‘BurnB!rd’ is passed to addDigit as the argument. None of the letters ‘o’, ‘e’, or ‘l’ exist in this string so the boolean loop on lines 25-28 would evaluate to False. This would cause line 32 to execute rather than line 30, thus returning the concatenated value of password as ‘BurntB!rd69’.