Task 3 was simpler than I thought it was going to be. The overall task was to develop upon task 2 so the program could also reconstruct a paragraph, retrieving information from files that were previously created. Now, previously I mentioned that I don't think there is any point in creating the index file, and here's why. The index file is just a file with numbers. That's it. The numbers have no other value than themselves, so without some sort of dictionary to look up the numbers value, they would be useless in the assembly of the sentence. Each word that was stored in the words file was on its own line for this exact reason. So, that later when they are to be put together, it was obvious what order they needed to be in and minimal effort was required to do so by the program. Anyway, onto how I started task 3.

```
1  def readfile():                          #creates the function which will be used to read compressed files.
2      sentencelist = []                    #creates a blank array which will be added to later.
3      attempt = "no"                       #creates a variable witht he value of "no" to be used for a while loop
4      while attempt == "no":               #creates a loop so if an invalid input has been entered by the user, it loops back to the begining
5          filename = input("Enter file name or type cancel. If you used this program to create the file, the filename will be whatever you ca
6          if filename == "cancel":         #if the user enters cancel it takes them back to the starting screen
7              menu()                       #starting screen where user enters what they wish to do (read, write or compress)
8          elif filename == "Cancel":       #again the cancel option but with a captiol C to support case senstivity
9              menu()
```

Here you can see the first function which is the main feature of task 3, the 'decompression' and reconstruction of a sentence that was previously 'compressed' into 2 files. You're probably thinking, "huh, function?!". To well structure my task 3, I decided to split it up into 3 functions. A function is almost like a smarter variable. It stores a bunch of code which can later be used just buy stating the functions name. So rather than having to copy and paste code, you can just refer to the function name. Above you can see the first part of the readfile function. It just contains a blank list variable and a while statement which constantly runs the filename input. This is where the user will enter the name of the file they wish to decompress into a sentence. It will only find the file if it is in the same directory as the program is running in. You can see the two if statements basically saying if the user types cancel then return to the menu function. You'll see what that looks like later. It's funny because I am literally thinking right now why I didn't just turn the users input into lowercase so I only needed one if statement. I guess I'll add that in a sec.

```
10         try:                             #runs this peice of code within an error exception
11             with open(filename + '.txt', 'r') as f:         #opens the file that the user identified previously in the filename variable
12                 words = f.readlines()                       #sets the varaible words to have all the data in the file
13                 for word in words:                          #for each word in the file
14                     if word not in sentencelist:            #if any of the words it goes through is not in the empty array sentencelist
15                         word = word.strip('\n')             #removes the newline from the word to prevent logical errors
16                         sentencelist.append(word)           #adds the word to the previously empty array
17                         attempt = "yes"                     #ends the while loop so the code continues to the user being takent to the menu
18         except FileNotFoundError:                           #if this error occurs, do the following
19             print("Oops! Doesn't look like there's a file with that name (not locally anyway).")    #prints that a file was not found
20             attempt = "no"                                                                          #attempt remains "no" so it loops back
21     sentencestring = " ".join(sentencelist)                                                         #assigns a space character and the the
22     print("The following sentence has been constucted: \n \n" + sentencestring)                     #outputs the sentence that was put toge
23     menu()                                                                                          #returns user to main screen
```

Now here you can see something new. I managed to figure out how to add error exceptions! This really was a breakthrough because one error can stop the entire program from running. To do this you basically write "Try:" and the code block you want it to run underneath it. Then when you're done, you can write "except" and the error that you could come up followed by the response to the error which in this case is the "Oops, doesn't look like there's a file…". The code block under the try bit is the decompression part. What it does is open the file using the variable that the user previously entered and using a for loop, get each word in the file and add it to the blank sentence list called sentencelist. Oh, and I used the .strip() function to remove the newline that was added to each word. After that I ended the while loop to continue on where I used the .join() function to put

together the sentence using all the words in the sentence list. You can also see how if the variable that the user inputted is used for the filename but a file isn't found, it follows the error exception and outputs "file not found" etc.

```
25  def compressfile():                                                          #the following code will make the compr
26      sentence = input("Enter sentence to compress or type cancel \n > ")      #user enters the sentence they wish to
27      sentencename = input("Enter the sentences' name or type cancel \n > ")   #user enters the name they wish to assi
28      if sentence == "cancel" or sentencename == "cancel":                     #if the user enters cancel in either of
29          menu()
30      elif sentence == "Cancel" or sentencename == "Cancel":                   #added capitol C's to help non-case sen
31          menu()
32      sentencelist = sentence.split()                  #creates a list that contains each induvidual word from the sentence that the u
33      words = {}                                       #refer to task 2 from line 5 for information on this commentary
34      index = 0
35      for word in sentencelist:
36          if word not in words:
37              words[word] = index
38              index += 1
39
40      with open(sentencename + 'index.txt', 'w') as f:
41          for word in sentencelist:
42              f.write('%d\n' % words[word])
43
44      with open(sentencename + 'words.txt', 'w') as f:
45          for word in sentencelist:
46              f.write(word)
47              f.write('\n')
48      print("File successfully compressed!")
49      menu()
```

Ok so this next part really doesn't need that much explaining as it's literally task 2 put into a function called compressfile. Because that is what task 2 was, compressing a file.

```
51  def menu():                                          #the following code will be the functioning for the main menu
52      print("Hello welcome to my controlled assessement task 3. What do you want to do? \n 1) Read file \n 2) Compress sentence into file \n
53      optionloop = int(0)                              #had to make sure the program new that this is an integer at 0.
54      while optionloop == 0:                           #while the option loop is 0 do the following code
55          try:
56              choice = int(input("> "))                #user enters the number choice they wish to do
57              while choice not in (1, 2, 3):           #another while loop so if the user enters an invalid option it loops them to li
58                  choice = int(input("Enter a valid choice (1 to 3). \n >  "))
59              if choice == 1:                          #if the user enters 1 it take them to readfile function
60                  optionloop == 1                      #ends the while loop allowing the code to continue to the correct function
61                  readfile()
62              elif choice == 2:
63                  optionloop == 1
64                  compressfile()
65              elif choice == 3:
66                  optionloop == 1
67                  print("Cya!")                        #a goodbye message for the user if they decide to quit the program
68                  exit                                 #ends the program
69          except ValueError:                           #if the program recieves the value error (can be caused by using a string in th
70              optionloop == 0
71  menu()                                               #the first peace of code the program actually runs is this. Shows the user the
```

This part is the menu function which is the bridge between all the functions. All it does is print to the user the options that they have, so compress, decompress or quit, and then it takes them to the function. It really is quite simple. It also contains an error exception because if the user entered a string (words) it would go crazy and tell me there was a value error since I declared the input as an integer. You can see at the bottom I have put menu(). This just runs the function when the program start. In matter of fact, it's actually the first line that is executed by python as functions are ignored

unless called upon. So there you have it. Task 3 over and done with. I'll be sure to add that lower case thing on all the inputs to prevent case sensitivity on the if statements but other than that I'm pretty happy with it. I sure as hell hope the examiner is too...