

```
... @@ -0,0 +1,18 @@
1 +sentence = input("Enter sentence to compress: \n>")
2 +sentencelist = sentence.split()
```

For task 2 there are clear stages for the code. The first of which you can see above, is to input the sentence and then store each word in a separate list. This was easily done as previously done.

```
3 +words = {}
4 +index = 0
5 +for word in sentencelist:
6 +     if word not in words:
7 +         words[word] = index
8 +         index += 1
```

The second job to do was to take every word from the sentencelist and put them into another list called words but each word would have the value of its index. Now, I know that probably doesn't really make sense so here's an example: Let's say that sentencelist has the words ('hello', 'my', 'name', 'is', 'cameron', 'name'). From here, the word 'hello' will be added to the words list but along with its index which is 0, this would look like {'hello': 0}. So, my final words list would look like this:

```
{'hello': 0, 'my': 1, 'name': 2, 'is': 3, 'cameron': 4}
```

For what purpose is this for you ask? It's so later on, I can write these indexes to a file in the correct order. However I must confess, I think that it's actually pointless. I'll tell you why later on as it will make more sense. Now onto outputting information into files. Two files need to be made, an index file which contains the indexes of the words and a words list which, obviously, contains the words from the sentence. Writing the indexes to a text document was simple. First create and then open index.txt. Second, using a simple for loop, for each word in the sentence list write its index value (which is found in the words list) into the text file. You can see that for each thing written, a new line is written with it so each index has its own line.

```
10 +with open('index.txt', 'w') as f:
11 +     for word in sentencelist:
12 +         f.write('%d\n' % words[word])
```

As for writing the words to another text document, it was really easy. Again, create and open a words.txt file and, with the help of a really simple for loop, take each word from the sentence list and write it to the text document. Again, each word has its own line so it can easily be recognised as an individual word later on. Then of course a nice message to say all the hard work has been done. Oh and in case you're wondering, the text documents are stored in the same directory as the program was run, so I would recommend running the program in its own folder.

```
14 +with open('words.txt', 'w') as f:
15 +     for word in sentencelist:
16 +         f.write(word)
17 +         f.write('\n')
18 +print("File compressed into indexes and words.")
```

Later on I noticed two lines which could be reduced into one through string formatting. I simply made it so it writes the new line and the word at the same time instead of separately, hopefully making the program run faster.

```
14 -with open('words.txt', 'w') as f:
15 -     for word in sentencelist:
16 -         f.write(word)
17 -         f.write('\n')
18 -print("File compressed into indexes and words.")

14 +with open('words.txt', 'w') as f: #opens or creates a file called words
15 +     for word in sentencelist:#for each item in sentencelist
16 +         f.write(word + '\n') #write the item onto the text document along with a newline for the next item
17 +print("File compressed into indexes and words.") #informs the user that the compressing process has completed
```

And here's the final version of task two!

```
1 sentence = input("Enter sentence to compress: \n>") #user enters a sentence
2 sentencelist = sentence.split() #sentence is split into individual words and then stored in a list called sentencelist
3 words = {} #create a blank array for later
4 index = 0 #set the index value at 0 to start
5 for word in sentencelist: #goes through each item in the sentencelist
6     if word not in words: #if the item isn't in the words array
7         words[word] = index #the item in the words arrays index is changed to the appropriate position
8         index += 1 #the index variable is increased by one, so the next item found can have its index set to the second position
9
10 with open('index.txt', 'w') as f: #opens or creates a file called index
11     for word in sentencelist: #for each item in the sentencelist
12         f.write('%d\n' % words[word]) #write the items position number onto the text document along with a newline for the next item
13
14 with open('words.txt', 'w') as f: #opens or creates a file called words
15     for word in sentencelist: #for each item in sentencelist
16         f.write(word + '\n') #write the item onto the text document along with a newline for the next item
17 print("File compressed into indexes and words.") #informs the user that the compressing process has completed
```